VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Užití shlukovacích algoritmů v sociálních sítích

Using of the Clustering Algorithms in the Social Networks

2013 Bc. Jakub Hromada

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Jakub Hromada**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Užití shlukovacích algoritmů v sociálních sítích
Using of the Clustering Algorithms in the Social Networks

Zásady pro vypracování:

Cílem práce je přehled shlukovacích algoritmů často používaných při analýze sociálních sítí a implementace minimálně dvou odlišných shlukovacích algoritmů.

1. Sociální sítě - teoretický úvod a popis odlišností sociálních sítí od jiných síťových struktur.
2. Shlukovací algoritmy, které jsou vhodné pro sociální sítě. Jejich popis a charakteristika použití.
3. Implementace minimálně dvou vybraných algoritmů.
4. Experimenty s algoritmy nad různými datovými kolekcemi, vhodná vizualizace a vyhodnocení výsledků.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.
Networks: An Introduction. M.E.J. Newman
Computational Social Network Analysis: Trends, Tools and Research Advances. Ajith Abraham, Aboul-Ella Hassanien, Vaclav Snásel

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Pavla Dráždilová**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.*

V Ostravě 5. května 2013 .....Hromada.....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 5. května 2013 .....Hromada.....

Rád bych tímto poděkoval Mgr. Pavle Dráždilové, Ph.D. za odborné vedení diplomové práce a čas, který mi věnovala při zpracování diplomové práce.

Abstrakt

Tato diplomová práce se zabývá shlukovacími algoritmy a jejich použitím v sociálních sítích. Cílem práce bylo prověřit několik shlukovacích algoritmů nad různými datovými kolekcemi a výsledky analyzovat. Vybrány byly tři algoritmy. Jedná se o K-means, Fuzzy C-means a Markov Cluster algorithm. Tyto algoritmy byly implementovány a jejich funkcionalita byla vyzkoušena na sérii několika datových sad. Výsledky byly následně vyhodnoceny a podrobeny analýze.

Klíčová slova: shluk, shlukování, sociální sítě, K-means, Fuzzy C-means, Markov Cluster algorithm, MCL, analýza dat

Abstract

This thesis deals with coalescing algorithms and their using in social networks. The aim of the thesis was to check several clustering algorithms with different data collections, and analyse the results. In my work was used three algorithms, namely: K-means, Fuzzy C-means and Markov Cluster algorithm. These algorithms were implemented and their functionality was tested on a series of multiple data sets. The results were evaluated and analysed.

Keywords: cluster, clustering, social network, K-means, Fuzzy C-means, Markov Cluster algorithm, MCL, data analysis

Seznam použitých zkratek a symbolů

apod.	–	a podobně
atd.	–	a tak dále
FCM	–	Fuzzy C-means
HTML	–	Hyper Text Markup Language
MCL	–	Markov Cluster Algorithm
MySQL	–	Markov Cluster Algorithm
např.	–	například
PHP	–	Hypertext Preprocessor
phpBB	–	PHP Bulletin Board
tzv.	–	takzvaný
URL	–	Uniform Resource Locator
WWW	–	World Wide Web

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

V současné době můžeme čím dál tím více slyšet o fenoménu sociálních sítí. Prakticky neuplyne den, kdy by nebyla nějaká zpráva v televizi, rádiu, v novinách a samozřejmě na internetu, týkající se některé ze sociálních sítí. I když se jedná o poměrně novou metodu komunikace mezi lidmi, velmi rychle se rozšířila a s rostoucí popularitou internetu se nadále rozvíjí.

Pod pojmem sociální síť se nerozumí jen komunita lidí, kteří spolu komunikují přes web. Je to velice široký pojem, který by ve své nejjednodušší formulaci mohl znít jako společenství navzájem komunikujících či jinak interagujících lidí. Tedy i obyčejná rodina tvoří malou sociální síť. Maminka s tatínkem mají dvě děti, sourozence, rodiče, příbuzné, kamarády atd. Každý má s někým určitý vztah. Tato malá sociální síť by se dala reprezentovat grafem, kdy každý člen rodiny by tvořil vrchol a hrana by představovala vztah mezi nimi. Na první pohled by bylo patrné, kdo z rodiny je černá ovce a straní se kolektivu nebo naopak, kdo je středem pozornosti. Tomuto rozboru se odborně říká analýza sociálních sítí a více se o tomto tématu dozvíte v kapitole druhé s názvem Sociální sítě. Dále jsou zde taktéž zmíněny způsoby reprezentace sociálních sítí nebo třeba rozdíly mezi sociálními sítěmi a jinými síťovými strukturami.

Jedním z hlavních nástrojů při analýze sociálních sítí jsou shlukovací algoritmy. Jedná se o speciálně navržené algoritmy, které mají za úkol vytvořit shluky dat, jež jsou si podobné. Využití najdou všude tam, kde se hledají společné znaky či vlastnosti. Věda je používá při analýze proteinů, prohledávání DNA nebo například při hledání plagiátů. Čím dál větší roli hrají také v oblasti marketingu, kde pomáhají nalézat zboží, které by se k danému zákazníkovi nejvíce hodilo. Na základě sběru dat, které na sebe uživatel sám nechtěně prozradí, se provádějí různé statistiky, analýzy a firmy se snaží nalézt co nejoptimálnější reklamu pro svého potenciálního zákazníka. Použití shlukovacích algoritmů je opravdu velmi široké.

Cílem této diplomové práce bylo nalézt a vyzkoušet několik shlukovacích algoritmů, jež by byly vhodné pro analýzu sociálních sítí. Pro vektorová data jsme vybrali algoritmus K-means a Fuzzy C-means. Oba algoritmy pracují na podobném principu, ale vrací odlišně interpretovatelné výsledky. Na data, která předem tvořila graf, jsme zvolili algoritmus Markov Cluster algorithm. Tento algoritmus prochází graf a odstraňuje slabé hrany, čímž v konečném důsledku vzniknou samostatné shluky dat. Více se popisu jednotlivých algoritmů věnujeme v kapitole třetí.

Pro kolekci vektorových dat byla použita databáze uživatelů fóra o noteboocích. Extrakcí dat z tohoto fóra se zabývá kapitola čtvrtá. Je zde popsána funkčnost a způsob provedení programu, který data stahoval. Získaná data bylo poté nutné přetransformovat do podoby vhodné pro zmíněné vektorové shlukovací algoritmy. Pro algoritmy pracující s grafem byla použita data veřejně dostupná z internetu.

V páté kapitole se tato práce věnuje samotné implementaci vybraných shlukovacích algoritmů. Pomocí diagramu tříd jsou názorně vysvětleny principy naší aplikace a jsou zde také zmíněny problémy, které při implementaci vznikly. Dále se zde seznámíme s naim-

plementovanou aplikací a vysvětlíme si její základní funkčnost a nastavení. Na konci kapitoly je popsána implementace modularity, pomocí níž měříme kvalitu shluků v grafu.

Předposlední kapitola obsahuje experimenty s naprogramovanými algoritmy. Porovnává se kvalita shlukování pomocí modularity, případně optimální počet shluků nebo počet iterací algoritmu K-Means. Vypočítaná modularita se porovnává s modularitou v programu Gephi, který zároveň slouží i jako vizualizační nástroj. Naměřené výsledky jsou analyzovány a následně vyhodnoceny.

Závěrečnému zhodnocení této diplomové práce se věnujeme v poslední kapitole. Shrnujeme zde své poznatky, které jsme nabyli při práci se shlukovacími algoritmy.

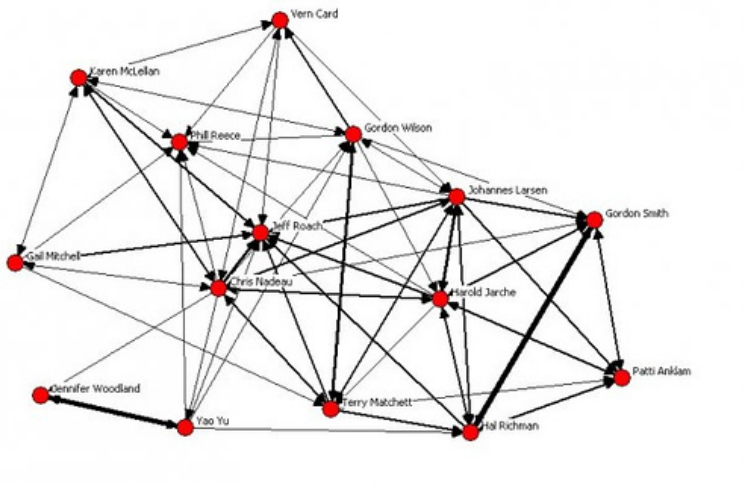
2 Sociální sítě

V této kapitole bude nastíněna problematika sociálních sítí. Úvod této kapitoly je věnován definici sociální sítě. Jsou zde rozebrány různé druhy sociálních sítí a jejich vlastnosti. Najdeme zde také zmínku o historii sociálních sítí a přehled nejdůležitějších sociálních sítí dneška. Další podkapitoly budou věnovány teorii grafu. Rozebírá se zde rozdíl mezi sociálními sítěmi a jinými síťovými strukturami. Jedná se hlavně o popis náhodných grafů, bezškálových sítí a regulárních sítí.

2.1 Definice sociálních sítí

Sociální struktury jsou součástí všech sociálních vztahů člověka k člověku. Podle Nadala je sociální struktura sítí sociálního vztahu, jež je vytvořena mezi lidmi, kteří na sebe vzájemně podle svých vzorců chování reagují. Karl Mannheim považuje sociální struktury za abstraktní a nehmotný fenomén. Jednotlivci jsou vzájemně propojeni v určitém uspořádání a vytvoří tak vzor sociální struktury. Z uvedených definic tedy plyne, že sociální struktura zachycuje to, co je i přes možné změny poměrně stálé a čím jsou jednotlivé společnosti od sebe odlišovány. Jinak řečeno jedná se o stabilní charakteristiku určitého sociálního systému [19].

Newman definuje sociální síť jako soubor lidí nebo skupin lidí ve struktuře kontaktů nebo jako interakce mezi těmito jednotlivci. Vztahy mezi jednotlivci mohou odrážet jejich vzájemné přátelství, obchodní vztahy mezi společnostmi, sňatky mezi rodinami apod. [27]. Sociální síť můžeme znázornit graficky, viz obrázek 1.



Obrázek 1: Příklad grafického znázornění sociální sítě [29]

S rozvojem internetu vzniká řada služeb, díky nimž byl umožněn rozvoj sociálních sítí i v prostředí internetu. Internetová sociální síť je webovou službou, která byla primárně založena za účelem sdružování uživatelů. Skrze tyto služby jim byla umožněna

jednoduchá forma komunikace a také možnost sdílení různých informací. Mohou být sdíleny obrázky, povídky, fotografie, vlastní nápady nebo kratičké zprávy, to vše v závislosti na druhu sociálního média. Vznik prvních sociálních sítí v prostředí internetu se datuje již u kořenů samotného internetu. Zprvu byla uživatelům umožněna velmi jednoduchá webová prezentace včetně možnosti posílání zpráv. Dalším krokem v rozvoji sociálních sítí v internetovém prostředí byl vznik blogů. První velký boom však nastal až v letech 2002 – 2004 a toto vysoké tempo rozvoje trvá dodnes [31].

2.2 Analýza sociálních sítí

Do analýzy sociálních sítí jsou zahrnuty metody, skrze něž je proveden rozbor sociální sítě z hlediska vazeb jedinců. Analytiky, kteří provádí analýzu sociálních sítí, je možné rozdělit do dvou skupin. Do první skupiny lze zařadit takové analýzy, které budou spíše analyzovat sítě ze sociologického hlediska, především hodnocení sociálních vazeb mezi jednotlivci a jejich vzájemný vliv. Druhá skupina bude využívat matematické postupy a pro znázornění síťových procesů bude využívána např. teorie grafu.

Nejčastější zobrazení sociální sítě je v podobě grafu nebo incidenční matice. V případě grafu je síť tvořena sociálními vazbami, kdy uzly znázorňují jedince a hrany představují vztahy mezi těmito jedinci. Hlavními vlastnostmi, které jsou v rámci analýzy sociálních sítí zjišťovány, je celková velikost sociální sítě, struktura, obsah a složení [5].

Obrázek 2: Webová sociální síť [15]

2.3 Historie sociálních sítí

Když pomineme online sociální sítě, tak historie sociálních sítí sahá až daleko do minulosti. Díky lidské vlastnosti se shlukovat do skupin, lidé už od nepaměti tvořili sociální sítě. V podstatě každá skupinka dvou a více lidí se dá považovat za sociální síť. Nicméně z vědeckého hlediska nejsou takové sítě užitečné, protože nemáme žádné údaje nebo data,

které bychom mohli zpracovávat. To se však začíná měnit s rozmachem počítačů a hlavně internetu.

2.3.1 Internetová diskuze - fórum

Internetová fóra (diskuze) jsou webové stránky nebo části webové stránky, kde probíhá on-line sdílení informací a myšlenek skrze internet. Tím je umožněno mnoha lidem po celém světě diskutovat o svých znalostech, zkušenostech, odborných znalostech atp. Online fórum je komunikační médium, díky němuž je zprostředkována komunikace mezi více uživateli. Obvykle tato komunikace probíhá prostřednictvím psaní příspěvků, ale existují i jiné techniky, jako je video či hlasové konference nebo chatování. Většinou mohou obsah fóra zobrazit i anonymní návštěvníci, ale psaní zpráv a vytváření nových témat je umožněno pouze registrovaným uživatelům. Fórum je další možností uživatelů, jak komunikovat skrze internet o všech otázkách života. Díky fóru je také možné shromažďovat informace od všech uživatelů na různá témata, která mohou mít následné využití např. pro obchodní účely [33].

Fóra jsou členěna na jednoduchá (v případě takového fóra jsou příspěvky řazeny chronologicky za sebou) nebo strukturovaná, které jsou tvořeny vlákny. Vlákna jsou definována jako jednotlivé rozhovory v rámci internetového fóra (anglicky Thread). Tyto vlákna obsahují příspěvky členů diskuze, které na sebe vzájemně navazují, v závislosti na tom, na který příspěvek původně reagují.

V prostředí internetových fór mají uživatelé možnost například zobrazení kompletní diskuze, nastavení počtu zobrazených příspěvků, tisk atd. V rámci fóra jsou jednotlivé diskuze členěny dle témat, kdy jednotlivá témata jsou kontrolována moderátory a správci, kteří mohou přesouvat jednotlivá vlákna, mazat apod. [18].

Fóra jsou členěna na:

- Diskuze ke stránce nebo článku: taková forma diskuze je doplňková funkce, díky které je uživatelům k dispozici možnost zpětné reakce na články, produkty či služby, ke které je tato diskuze vztažena.
- Návštěvní kniha (guestbook): skrze guestbook uživatelé píší komentáře příp. jiné zprávy určené pro danou webovou stránku. Je tak možné získat zpětnou reakci svých návštěvníků. Navíc mají i ostatní návštěvníci možnost vidět toto hodnocení.
- Diskuzní fórum: je charakterizováno jako místo, kde je uživatelům nabídnuta možnost začít komunikaci ve formě vlákna a také odpovídat na příspěvky ostatních uživatelů. Jedná se o rozsáhlý diskuzní projekt (server), kde je obsaženo množství témat (stránek). V každém fóru může být obsažen velký počet témat a podtémat, obsahující velké množství vláken.
- Otázky a odpovědi: na webových stránkách je uživatelům k dispozici veřejné zodpovídání dotazů, kdy je uživatelem zaslán dotaz na majitele webových stránek a následně je přidána kvalifikovaná odpověď z řad zástupců stránek. Dotazy jsou vztaženy k obsahu těchto stránek - články, nabízené produkty a služby apod. Tyto otázky a následné odpovědi jsou veřejně přístupné [18].

2.3.2 Facebook

Facebook je dnes největší online sociální sítí na světě. Původně však byla tato síť určena pouze studentům Harvardské univerzity. Facebook byl založen za účelem vzájemného poznávání studentů na univerzitě a stal se brzy velmi populárním. Rychle se rozšířil i do dalších vysokých škol v oblasti Bostonu a dalších, které patří do tzv. Ivy League. Za zrodem Facebooku stál student Harvardské univerzity Mark Zuckerberg a jeho kolegové Eduard Saverin, Dustin Moskovitz, Chris Hughes a Andrew McCollum. Původně byl založen pod doménou „thefacebook.com“, kdy název byl odvozen od papírových letáků nazývaných Facebookes, které jsou rozdávány všem prvákům na amerických univerzitách, za účelem bližšího seznámení studentů. Dále byla tato sociální síť rozšířena i mezi schválené zahraniční univerzity.

Nejprve bylo na Facebooku umožněno pouze srovnání dvou studentů a další lidé měli určit, který z nich je atraktivní a který ne. Postupem času bylo na Facebook přidáno mnoho funkcí. Nyní jsou uživatelům k dispozici novinky přátel, větší ochrana osobních údajů, uživatelé přidávají obrázky, komentáře, píší si zprávy, vytváří vlastní stránky a skupiny. Nové aplikace jsou přidávány každým dnem. Posláním Facebooku je, aby svět byl otevřenější a více propojený. Aby lidé využívali Facebook z důvodů udržení kontaktu s přáteli a rodinou, zjistili co je nového ve světě a zároveň mohli sdílet vlastní názory.

V prosinci 2007 měl Facebook 57 milionů aktivních členů a byl největším studentským webem dle počtu aktivních uživatelů. V hodnocení tak stoupl oproti roku 2006 ze 60. příčky na příčku 7. a stal se tak jednou z nejnavštěvovanějších webových stránek světa. V USA byl nejvíce užívanou webovou stránkou pro sdílení fotografií (za týden bylo sdíleno více než 60 milionů fotografií) [36].

Příjmy Facebooku plynou zejména z reklamy. Jedná se především o bannerovou reklamu, která se zobrazuje každému uživateli. Je optimalizována dle údajů, které má uživatel ve svém profilu vyplněny a v závislosti na chování daného uživatele ve Facebookovém prostředí. Hodnota společnosti byla odhadnuta v roce 2006 přibližně na 100 milionů USD.

Dle údajů Facebooku bylo v roce 2010 vytvořeno 610 milionů profilů. Každých 60 sekund je posláno 230 tisíc zpráv, aktualizováno 95 tisíc statusů, napsáno 80 tisíc statusů na zeď, sdíleno 65 tisíc fotografií, 50 tisíc odkazů a zároveň přibude půl milionu komentářů. Průměrný uživatel stráví přibližně více než 6 hodin na Facebooku za měsíc. Značky, které mají nejvíce fanoušků, jsou Coca-Cola (21,6 milionů fanoušků), StarBucks (19 milionů fanoušků) a Oreo (16,2 milionů fanoušků). Dle údajů Facebooku je součástí této sociální sítě více než 3,8 milionů uživatelů z České republiky. V roce 2012 měl Facebook více než 1 bilion uživatelů [34].

2.4 Úvod do teorie grafů

Teorii grafů řadíme mezi relativně mladou část matematiky. Kořeny teorie grafu sice sahají až do 18. století, ale první kniha věnována této teorii vyšla až v roce 1936. Pomocí teorie grafů jsou zkoumány vlastnosti struktur, které nazýváme grafy. Grafy jsou tvořeny uzly, které jsou vzájemně propojeny pomocí hran. Grafy jsou znázorňovány tedy jako množina bodů, které jsou vzájemně propojeny čárami.

Formálně lze zapsat graf uspořádanou dvojicí $G = (V, E)$, kde V je množina vrcholů a E množina hran propojených vrcholů. Hranu mezi vrcholy u a v píšeme jako $\{u, v\}$, nebo zkráceně uv. Vrcholy spojené hranou jsou sousední. Pro celkový počet vrcholů se někdy používá označení $|V|$ a $|E|$ znamená celkový počet hran.

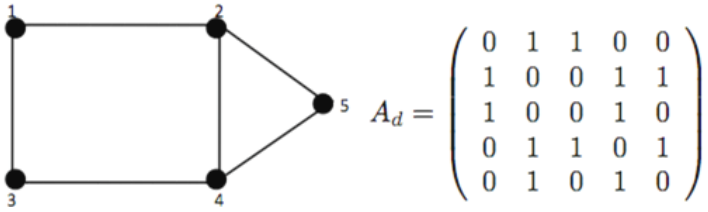
Velmi mnoho věcí, se kterými se dnes v reálném světě setkáváme, můžeme vyjádřit grafem jako matematickým modelem. Za příklad lze uvést např. železniční síť, kdy města představují uzly a železnice je znázorněna hranami, nebo třeba pracovní postup stavby mostu lze také vyjádřit grafickým modelem, kdy hrany představují jednotlivé dílčí pracovní činnosti, které na sebe bezprostředně navazují [30].

Graf je datová struktura, která se používá pro vizualizaci informací. Skládá se z objektů, jenž reprezentují vrcholy a vztahy mezi objekty jsou znázorněny pomocí hran. Rozlišujeme dva typy grafů:

- neorientované - hrany jsou dvouprvkové množiny,
- orientované - hrany jsou uspořádané dvojice a je pevně dána orientace.

Grafy mohou být znázorněny několika způsoby. Nejznámější je kreslený graf, kdy kolečka představují vrcholy a jsou pospojovány čárami, jenž symbolizují hrany. Tyto hrany mohou být ohodnocené. Pokud hrana váhu obsahuje, jedná se o přidanou informaci. Například na grafu silnic znamená váha hrany délku silnice.

Dalším způsobem znázornění grafu je pomocí matice sousednosti. Jedná se o čtvercovou matici $n \times n$, kde hodnota jednotlivých prvků matice a_{ij} je rovna počtu (váze) hran vedoucích z vrcholu i do vrcholu j [1]. Příklad si můžete prohlédnout na obrázku 3.



Obrázek 3: Graf a matice sousednosti [8]

V naší práci budeme používat i pojem stochastická matice. Stochastická matice je čtvercová matice s nezápornými prvky, jejichž součet v každém řádku je roven 1 [16].

Definice 2.1 *Čtvercová matice $S = (s_{ij})$ řádu n se nazývá (řádkově) stochastická, platí-li*

$$s_{ij} \geq 0, \qquad \sum_j s_{ij} = 1, \qquad i, j = 1, ..., n$$

2.4.1 Sled, tah, cesta

Sled je na sebe navazující posloupnost hran, kdy vždy dvě za sebou následující hrany mají společný koncový uzel.

Tah mezi uzly u a v je sled mezi těmito uzly, ve kterém se žádná hrana nevyskytuje vícekrát.

Cesta mezi uzly u a v je tah mezi těmito uzly, ve kterém se žádný jeho vnitřní vrchol nevyskytuje vícekrát [27].

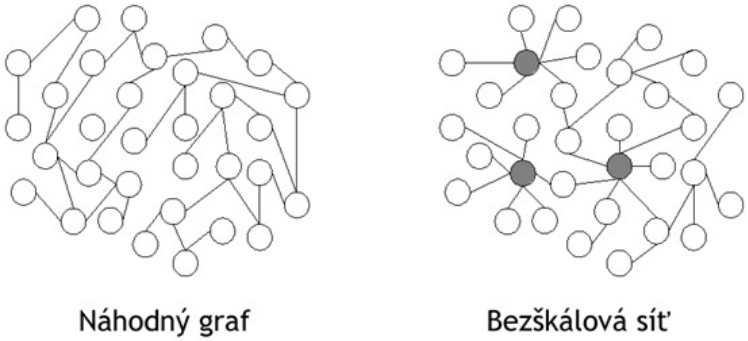
2.5 Náhodné grafy

Nejjednodušším modelem sítě je náhodný graf. Tento typ grafu byl poprvé popsán Paulem Erdösem a Alfrédem Rényim. Tito matematici předpokládali, že v tomto modelu jsou neorientované hrany umístěny náhodně mezi pevný počet uzlů n, kdy vrcholy vytvoří síť, v níž každá z $\frac{1}{2}n(n-1)$ je nezávisle přítomna s určitou pravděpodobností p a počet hran, který spojuje uzly, je rozdělen podle binomického nebo Gaussova rozložení, které je obvyklé pro náhodné grafy. Náhodný graf je tedy tvořen postupným přidáváním hran vždy mezi dva náhodné uzly, které zatím nejsou spojeny. Postupem času se ukázalo, že struktury a sítě se neřídí Gaussovým rozdělením a nemají náhodný charakter. Například World Wide Web nelze vyjádřit náhodným grafem, neboť webové stránky představující uzly jsou spojeny odkazy, které jsou orientovány (na jednotlivé webové stránky povedou odkazy z jiných webových stránek) [4].

2.6 Bezškálové sítě

Z důvodu, že ne všechny struktury je možné znázornit náhodným grafem, jsou využívány bezškálové sítě. V případě bezškálové sítě, neexistuje žádná hodnota, kolem které by byly prvky rozděleny, jako v případě náhodného grafu a Gaussova rozložení. Tento model také předpokládá, že na úplném začátku je stanoven přesný počet uzlů, mezi nimiž jsou vytvářeny vazby. Z tohoto důvodu jsou bezškálové sítě více reálné. Tyto sítě jsou definovány vztahem $P(k) = k^{\gamma}$, kde $P(k)$ vyjadřuje pravděpodobnost, že daný uzel sousedí s k dalšími uzly a γ je koeficientem distribuce $\gamma > 1$. Typickou vlastností pro bezškálové sítě jsou centra. Z tohoto centra vychází nejvíce hran. Díky těmto centrům, jsou bezškálové sítě velmi odolné struktury vůči odstraňování uzlů. Při náhodném odstranění uzlů, je velmi malá pravděpodobnost, že bude odstraněno centrum. V případě, že by centrum odstraněno bylo, dochází k rozpadu celé sítě. Tento model následně rozvinul Barabási a Albert, kdy byl popsán vznik bezškálové sítě z hlediska dvou principů:

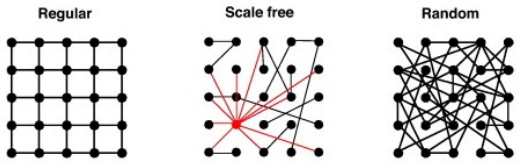
- Růst – v časovém intervalu jsou přidávány v rámci sítě nové uzly. Tyto sítě jsou vytvářeny dynamicky.
- Preferenční připojování – nový uzel je přidán k uzlům, které již existují, a je navázán x vazbami. Pravděpodobnost výběru uzlu je přímo úměrná celkovému počtu vazeb uzlu, ke kterému je nový uzel připojován. Tento model rozšířil původní model, avšak není zde řešena otázka rušení uzlů a vzájemných vazeb ani změna případně přidávání vazeb k uzlům, které již existují [28].



Obrázek 4: Příklad náhodného grafu a bezškálové sítě [7]

2.7 Regulární sítě

Regulární síť je síť s pravidelným grafem ve struktuře (tento graf se může opakovat). Vzhledem k této pravidelnosti vykazuje síť nízkou nebo nulovou entropii. Je to protějšek k náhodným sítím. V této síti je každý uzel dosažitelný z jiného uzlu v relativně malém počtu přechodů. Tyto sítě jsou zpravidla řídké, mají relativně malý průměr a malou průměrnou délku cesty. Výpočet centra, které je důležité pro tento druh sítě, je komplikované vzhledem k možnosti více centrálních bodů. Centrální bod je definován jako minimální cesta přes všechny uzly [21].



Obrázek 5: Síťové struktury [11]

2.8 Sociální sítě

Sociální síť je síťová struktura skládající se z množiny objektů, jejichž členové jsou mezi sebou propojeni jednou nebo více vazbami. Změnou vazeb v síti se změní i síť, a to i přesto, že všechny objekty zůstaly stejné.

Většina sociálních sítí vytváří ve své struktuře komunity. Jedná se o skupiny vrcholů, které mají mezi sebou vyšší hustotu hran než je hustota hran mezi shluky. Shlukem se dají charakterizovat společné zájmy, věk, povolání atd. Struktura sociálních sítí se liší případ od případu. Mohou tvořit velmi izolované struktury sítě, kde mají mezi sebou objekty jen pár vazeb až po velmi strukturované sítě, v nichž je většina objektů navzájem propojena [20].

3 Shlukovací algoritmy

Základním účelem shlukování (clustering), je najít shluk (komunitu) objektů, které mají navzájem více společných znaků. Jelikož není předem jasné, s jakými daty budeme pracovat, je obvykle nutné je pro shlukovací algoritmy předzpracovat. Dalším problémem je výběr správného shlukovacího algoritmu pro danou úlohu. Výběr závisí jednak na velikosti dat a potom zejména na druhu úkolu. Pro kompresi dat bude vhodný jiný algoritmus, než pro analýzu sociální sítě [14].

3.1 Shluková analýza dat

Pod pojmem shluková analýza dat se skrývá celá řada metod a přístupů, které mají podobný cíl, a to nalézt skupiny objektů vzájemně si podobných. Je využívána především tam, kde přirozenou tendencí objektů je vzájemně se seskupovat do přirozených shluků [17].

Definice 3.1 *Základním cílem shlukové analýzy je zařadit objekty do skupin (shluků), a to především tak, aby dva objekty stejného shluku si byly více podobné, než dva objekty z různých shluků [32].*

Jak z předešlé definice vyplývá, základním cílem shlukové analýzy je rozdělení prvků do shluků. Tento základní cíl je možné rozdělit na tří dílčí cíle [24]:

- Identifikace vztahu - pomocí nalezení shluků je možné objasnit strukturu mezi objekty a je poté snadnější odhalit vztahy, které se mezi jednotlivými objekty nachází.
- Zjednodušení dat - díky shlukové analýze je zjednodušen pohled na jednotlivé objekty.
- Popis systematiky - shluková analýza je využívána pro průzkumové cíle a taxonomii, neboli empirickou klasifikaci objektů.

Ve shlukové analýze nedochází k rozlišení atributů objektů na významné a nevýznamné. Jsou zde pouze odlišeny shluky. Pokud jsou atributy nesprávně zařazeny, může to být v důsledku zahrnutí odlehlých objektů, které mohou zkreslit výsledky této analýzy [24].

Vstupem pro analýzu dat je datová matice a výstupem shlukové analýzy identifikace shluků. Shluková analýza dat se zaměřuje na zkoumání podobnosti či nepodobnosti jednotlivých objektů [32].

3.2 Objekty a vlastnosti

Často bývá v odborné literatuře popsána jedna a tatáž věc více způsoby. Například objekt, prvek, záznam, datový bod je stále jedno a to samé. V této diplomové práci budeme jednotně používat slovo objekt. Pro každý objekt v n-dimenzionálním prostoru definujeme jeho vlastnosti neboli atributy. Objekt je vektor, skládající se z atributů.

Matematicky můžeme tento vztah vyjádřit následujícím způsobem. Datovou kolekci s n objekty, které jsou definovány d atributy můžeme zapsat $D = \{x_1, x_2, ..., x_n\}$, kde $x_i = (x_{i1}, x_{i2}, ..., x_{id})^T$je vektor popisující objekt i a x_{ij} je skalár popisující atribut j objektu x_i. Počtu atributů objektu se říká dimenze [12].

3.3 Vzdálenost a podobnost

Ve shlukové analýze hraje důležitou roli vzdálenost nebo podobnost. Jsou používány k popisu jak moc si jsou dva objekty podobné či nepodobné.

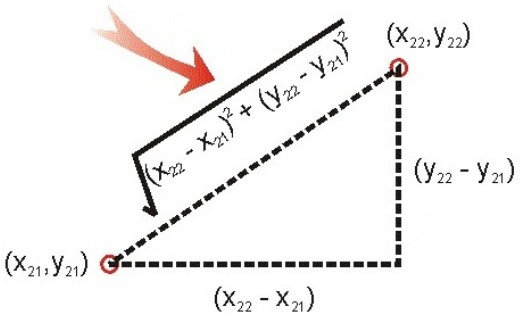
U vzdálenosti neboli nepodobnosti vyšší hodnota znamená větší nepodobnost mezi objekty či shluky. Typickým představitelem je Euklidovská vzdálenost, viz kapitola 3.3.1.

Naopak u podobnosti znamená vyšší číslo, větší podobnost objektů. Podobnost S lze převést na vzdálenost D pomocí vztahu $D = 1 - S$, přičemž ale nebude zachována trojúhelníková nerovnost. Opačným způsobem lze převést i vzdálenost na podobnost $S = 1 - D$, ale vzdálenost musí být normalizována. Představiteli podobnosti jsou v této práci kosinova podobnost a Jaccardova podobnost (Tanimotova podobnost) [12].

V této diplomové práci budeme používat vektorové shlukovací algoritmy K-means a Fuzzy C-means. Tyto algoritmy pracují se vzdálenostní metrikou. Proto když budeme používat některého z představitelů podobnosti, myslíme tím, že tuto podobnost převedeme na vzdálenost.

3.3.1 Euklidovská vzdálenost

Euklidovská vzdálenost (neboli geometrická metrika) je vyjádřena přeponou pravoúhlého trojúhelníku a výpočet je založen na Pythagorově větě [24].

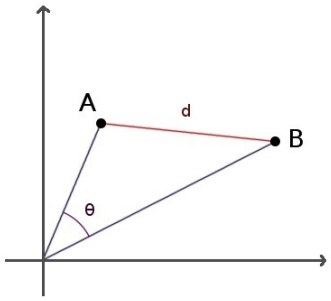


Obrázek 6: Euklidovská vzdálenost [13]

$$E(A, B) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

3.3.2 Kosinova podobnost

Nejznámějším představitelem podobnosti je kosinova podobnost. Představuje míru podobnosti dvou vektorů, která se získá následujícím výpočtem. Provedeme skalární součin vektorů a vydělíme jej součinem jejich absolutních hodnot. Tím získáme kosinus úhlu sevřeného oběma vektory.



Obrázek 7: Kosinova podobnost [26]

$$C(A,B) = \frac{\sum_{i=1}^{n} (a_i \cdot b_i)}{\sqrt{\sum_{i=1}^{n} (a_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (b_i)^2}}$$

3.3.3 Jaccardova podobnost

Jaccardova podobnost, v některé literatuře zvaná též jako Tanimotova podobnost, je dána vzorcem :

$$J(A,B) = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2 - \sum_{i=1}^{n} a_i \cdot b_i}$$

Udává podobnost mezi dvěma vektory (A, B). Čím více jsou si vektory podobné, tím více se výsledná podobnost blíží jedničce. Naopak při malé podobnosti bude hodnota klesat k nule [25].

Používá se k porovnávání otisků prstů a molekulových struktur. Podle některých zdrojů dává lepší výsledky než kosinova podobnost [22].

3.4 Rozdělení shlukovacích algoritmů

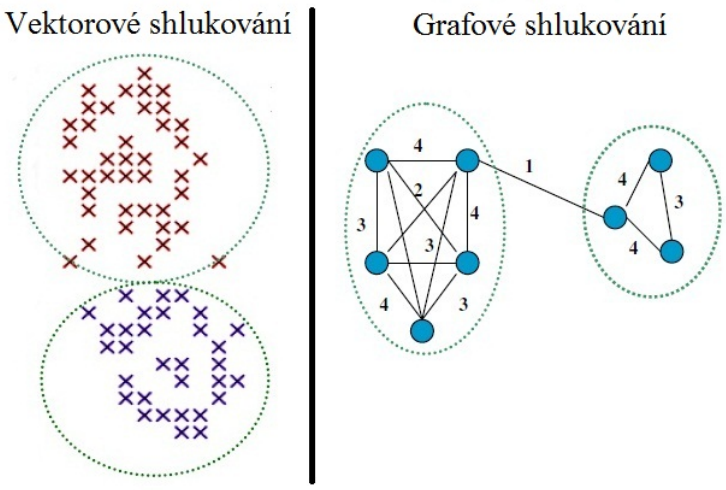
Existují dva druhy shlukovacích algoritmů, přičemž ale každý pracuje s jiným druhem dat [10].

1. Vektorové shlukování - data jsou definována pomocí vektorů a počítáme podobnost či vzdálenost mezi nimi. Podobné objekty jsou přiřazeny do shluků.

   Vektorové shlukování můžeme dále rozdělit na hard a soft shlukování.

   - Hard shlukování - tento druh shlukování přiřadí objekt právě do jednoho shluku. Mezi nejznámější představitele této skupiny můžeme zařadit algoritmus K-means.
   - Soft shlukování - se liší od hard shlukování tím, že každému objektu je přiřazena míra příslušnosti k jednotlivým shlukům. Objekt může patřit do více shluků, které se překrývají. Představitelem této skupiny je Fuzzy C-means [12].

2. Grafové shlukování - data jsou reprezentována grafem $G = (V, E)$, s množinou vrcholů a hran, které mohou být ohodnoceny a různým způsobem propojeny. Hlavní funkcí shlukování v grafu je odstranit slabé hrany, spojující shlukové struktury, čímž vzniknou samostatné shluky. Tento typ shlukování je v naší práci reprezentován algoritmem Markov Cluster algorithm (MCL) [10].



Obrázek 8: Rozdělení shlukovacích algoritmů [23]

V práci se budeme dále věnovat třem vybraným shlukovacím algoritmům. Jedná se o K-means, Fuzzy C-means a Markov Cluster algorithm. Tyto tři algoritmy byly vybrány proto, že se jedná o algoritmy s širokým využitím a v minulosti již byly použity i při analýze sociálních sítí.

3.5 K-means

Jedná se o nejznámější shlukovací algoritmus. Je založen na metodě nejbližších těžišť. Těžištěm myslíme střed shluku. Počet shluků k musí být znám ještě před spuštěním

algoritmu. Jsou-li těžiště a počet shluků předem známy, může být výpočet K-means postaven na nich. Avšak stanovit optimální počet shluků není úplně jednoduché. Jednou z možností je předběžná analýza dat, kdy zjistíme, s jakými daty pracujeme a stanovíme přibližný počet shluků.

Shlukovací algoritmus K-means pracuje s datovou kolekcí objektů, které jsou popsány n-dimenzionálními vektory, a pomocí metrik se počítá vzdálenost mezi nimi. Po počáteční náhodné inicializaci středů se těžiště přepočítává v několika iteracích. Do shluku jsou přiřazeny jen ty objekty, jejichž vzdálenost k těžišti je nejmenší. Výsledkem je, že každý objekt je přiřazen právě do jednoho shluku. Přepočet těžiště se získá průměrnou hodnotou všech objektů patřících do stejného shluku. Algoritmus končí, jakmile se težiště přestanou měnit a jejich hodnota zůstává stálá [24].

Hlavními výhodami algoritmu K-means jsou jednoduchá implementace, rychlost zpracování dat a možnost zpracovávat i velké datové kolekce. Mezi nevýhody se řadí náhodný výběr počtu shluků, citlivost na šum v datech (odlehlé hodnoty), které způsobí vychýlení středu shluku. Dalším problémem je počáteční výběr středů shluku. Tento výběr probíhá v klasické verzi K-means náhodně, což může způsobit, že se vyberou shluky, které jsou blízko sebe. Navíc pomocí náhodného výběru je obtížné získat stabilní výsledky měření.

Řešením by mohl být optimalizovaný výběr počátečních středů [37]. Počáteční střed je zde vybrán pomocí následujících kroků.

Algorithm 1 Optimalizovaný výběr počátečních středů

Input: počet shluků k
Output: seznam počátečních středů
1: náhodně vybereme objekt x_i
2: x_i uložíme do seznamu středů
3: pro všechny objekty ze seznamu středů hledáme nejvzdálenější objekt, který se stane dalším středem
4: opakujeme 3, dokud počet počátečních středů $\neq k$

Tento optimalizovaný algoritmus má za úkol vybrat počáteční středy takovým způsobem, aby vzdálenost mezi nimi byla co největší, avšak nastává zde problém s odlehlými objekty, které poté mohou zkreslit výsledky shlukování. Tato problematika bude popsána v kapitole s experimenty.

Výsledkem algoritmu K-means je, že každý objekt patří právě jen do jednoho shluku. Tomuto typu shlukování se říká hard. Odlišným typem je soft shlukování, které si popíšeme níže a jejímž představitelem v této práci je algoritmus Fuzzy C-means.

3.6 Fuzzy C-means

Fuzzy shlukování umožňuje přiřadit jeden objekt do více shluků. To je jeden z hlavních rozdílů oproti algoritmu K-means, ze kterého algoritmus Fuzzy C-means (FCM) vychází. FCM algoritmus se používá v mnoha oblastech vědeckého výzkumu pro svoji jednoduchost a robustnost [35].

FCM algoritmus mapuje konečnou množinu prvků $X = \{x_1, x_2, ..., x_n\}$ do kolekce fuzzy clusterů $C = \{v_1, v_2, ..., v_c\}$. Matice nepodobnosti $U = (u_{ij})$, kde $i = 1, 2, ..., c$ a $j = 1, 2, ..., n$. Potom u_{ij} označuje stupeň j-tého prvku k centru v_i.

Algoritmus pracuje ve 4 krocích [6]:

Algorithm 2 Fuzzy C-Means

Input: celé číslo c, udávající počet shluků (clusterů)

Output: míra přiřazení jednotlivých prvků ke shlukům

1: inicializujeme matici sousednosti U pomocí náhodného výběru středů

2: přepočítáme středy shluků dle vzorce

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}, 1 \leq i \leq c$$

3: aktualizujeme novou matici nepodobnosti U

$$u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{\|x_j - v_i\|^2}{\|x_j - v_k\|^2} \right)^{1/(m-1)} \right]^{-1} \quad (m > 1)$$

4: vypočítáme účelovou funkci

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m \|x_j - v_i\|^2$$

5: dáme $\epsilon > 0$, pokud $\left| J^{(n)} - J^{(n-1)} \right| < \epsilon$, algoritmus končí, jinak skoč na krok 2.

Výsledkem algoritmu je míra přiřazení všech objektů ke všem přepočítaným středům shluků. Tohoto jevu využíváme v experimentech, kdy se snažíme podle míry příslušnosti míchat barvy shluků takovým způsobem, aby překrytí barev graficky zvýraznilo příslušnost k jednotlivým shlukům.

3.7 Markov Cluster algorithm

Markov Cluster algorithm (MCL) je metoda pro nalezení shluků v grafu. Vyvinul ji Stijn van Dongen ve své disertační práci v roce 2000. Algoritmus pracuje s podobností mezi objekty, které jsou reprezentovány vrcholy grafu. Tato podobnost je dána jako ohodnocení hran mezi vrcholy. Metoda dále pracuje s Markovovou maticí přechodu dimenze $N \times N$ a je založena na principu náhodné procházky. Každý prvek matice C_{ij} je stupněm pravděpodobnosti přechodu uzlu i na uzel j. Hlavní chod metody zajišťují dvě operace nazvané Expand a Inflate [2].

Expand je číslo, které udává mocninu, na kterou budeme matici umocňovat. Obvyklé nastavení je expand $e = 2$.

Inflate značí také mocninu, nicméně je to mocnina, na kterou umocňujeme všechny prvky matice a následně matici normalizujeme. Operace inflace zvýhodňuje silnější hrany a slabší hrany zeslabuje. Matematicky lze tento postup definovat takto:

Definice 3.2 *Mějme matici $M \in \mathbb{R}^{k\times l}$, $M \geq 0$, a reálné kladné číslo r. Umocnění každého sloupce matice M mocninovým koeficientem r nazveme $\Gamma_r M$ a Γ_r bude inflační operátor s koeficientem r. Operace $\Gamma_r : \mathbb{R}^{k\times l} \to \mathbb{R}^{k\times l}$ je definována*

$$(\Gamma_r M)_{pq} = \frac{(M_{pq})^r}{\sum_{t=1}^{k} (M_{tq})^r}$$

Pokud není inflační index nastaven, autor v [10] jej doporučuje nastavit na $r = 2$.

Algoritmus pracuje v následujících krocích.

Algorithm 3 MCL

Input: orientovaný nebo neorientovaný graf, expanzní parametr e a inflační parametr r.
Output: prořezaný graf se shluky

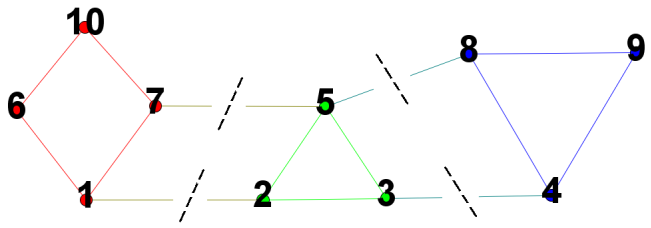
1: vytvoříme asociační matici
2: přidáme smyčku každému prvku na sebe samého
3: matici normujeme
4: umocníme matici na expanzní parametr e
5: provedeme operaci inflace uvedenou v definici 3.2
6: opakujeme kroky 5 a 6 dokud matice nezkonverguje
7: interpretujeme výsledky s nalezenými shluky

Přidání smyček na sebe samého je volitelná volba. Přidává se z důvodu, že sudé mocniny matice zkreslují výsledky u jednoduchých cest.

Algoritmus končí, když matice zkonverguje. Obvykle vznikne v každém sloupci jedno nebo několik čísel, jejichž součet je roven 1. Matice zkonverguje velmi rychle, obvykle stačí jen pár iterací algoritmu [10]. Na příkladu níže lze vidět původní matici sousednosti a výslednou zkonvergovanou matici. Na obrázku 9 můžeme vidět výsledný graf.

$$\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0
\end{pmatrix}$$

$$\begin{pmatrix}
1.0 & -- & -- & -- & -- & 1.0 & 1.0 & -- & -- & 1.0 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & 1.0 & 1.0 & -- & 1.0 & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.5 & -- & -- & -- & 0.5 & 0.5 & -- \\
-- & -- & -- & 0.5 & -- & -- & -- & 0.5 & 0.5 & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & --
\end{pmatrix}$$



Obrázek 9: Výsledek MCL shlukování

Při interpretaci výsledků je důležitým pojmem Attractor. Attractor má při konvergenci smyčku sám na sebe a „přitahuje“ k sobě všechny prvky patřící do stejného shluku. Interpretací výše uvedené matice vznikly tři shluky $\{1, 6, 7, 10\}$, $\{2, 3, 5\}$ a $\{4, 8, 9\}$.

3.8 Modularita

Modularita udává kvalitu rozdělení grafu neboli v našem případě kvalitu shlukování. Je založena na myšlence porovnávání hustoty hran uvnitř shluků s celkovým počtem hran. V této diplomové práci bude použita v kapitole 6, kdy budeme porovnávat různé nastavení shlukovacích algoritmů s vypočtenou modularitou.

Vstupem je neorientovaný graf $G = (V, E)$ s $n = |V|$ vrcholy a $m = |E|$ hranami. $|E(C)|$ je počet hran v clusteru (shluku) C. Shluk musí být nenulový, tj. musí obsahovat aspoň jeden vrchol. $\sum_{v \in C} deg(v)$ je suma stupňů vrcholů ve shluku C. Celkový počet hran v grafu označíme m [3].

$$q(C) = \sum_{C \in C} \left[\frac{|E(C)|}{m} - \left(\frac{\sum_{v \in C} deg(v)}{2m} \right)^2 \right]$$

Výsledkem je číslo $\langle 0, 1 \rangle$, kdy $q = 0$ znamená slabou kvalitu shluků. Znamená to, že počet bezkomunitních hran je příliš velký. Naopak $q = 1$ značí silnou strukturu uvnitř shluku. V praxi se ukazuje, že hodnoty modularity se pohybují většinou mezi 0,3-0,7 [9].

4 Extrakce diskuzního fóra

Tato kapitola bude pojednávat o extrakci dat z diskuzního fóra. Jedná se o veřejné diskuzní fórum se zaměřením na notebooky. Rozebereme si postup při zpracování, návrh databáze, extrakci dat s použitím jednoduchého programu a úpravu dat do souboru.

4.1 Internetové fórum notebooky-forum.notebook.cz

Jak je již z názvu patrné, jedná se o diskuzní fórum zaměřené na notebooky. Fórum spadá pod stránku www.notebook.cz. Provoz byl zahájen v květnu roku 2006 a jeho návštěvnost v posledních letech stále roste. Počet příspěvků k dnešnímu dni vyšplhal na slušných 310 tisíc a jeho průměrný denní přírůstek je 125 příspěvků. Na fóru je registrováno přes 22 tisíc uživatelů, ovšem hodně lidí zde chodí jen číst a ke čtení nepotřebují být zaregistrováni.

4.2 Seznámení s fórem

Fórum běží na phpBB systému, což je open source systém pro diskuzní fórum. Je vytvořený v PHP a využívá databázi MySQL či PostgreSQL.

Každý, kdo má zájem se zúčastnit diskuze na fóru, musí být zaregistrován. Registrace je samozřejmě zdarma, uživatel jen musí vyplnit několik údajů. Nejdůležitější z nich je login. Musí být unikátní pro celé fórum. Dále si uživatel může, ale není to povinností, vyplnit kontaktní údaje. U každého uživatele se eviduje celkový počet příspěvků.

Fórum je rozděleno do několika tematicky rozdílných kategorií. Nejoblíbenější kategorií je výběr notebooku, do které přispívá suverénně nejvíce uživatelů. Dále jsou také oblíbené kategorie týkající se jednotlivých značek notebooku. Zde se podle počtu příspěvků dá vydedukovat jednak oblíbenost značek notebooku nebo naopak přílišná poruchovost. Z ostatních kategorií můžeme dále uvést kategorie týkající se hardwaru, operačních systémů, her či inzerci.

Každá kategorie obsahuje různá témata, týkající se dané kategorie. Téma zakládá registrovaný uživatel a snaží se pomocí vhodného názvu přilákat další diskutující, kteří by mu mohli pomoci s jeho problémem. Někdy se nemusí jednat ani o problém, stačí navrhnout téma a diskutující začnou psát své názory.

Pokud chce uživatel reagovat na některé téma, musí napsat příspěvek. Každý příspěvek se skládá z textu či obrázku a dále obsahuje login autora příspěvku, datum založení příspěvku a identifikační číslo tématu, do kterého patří.

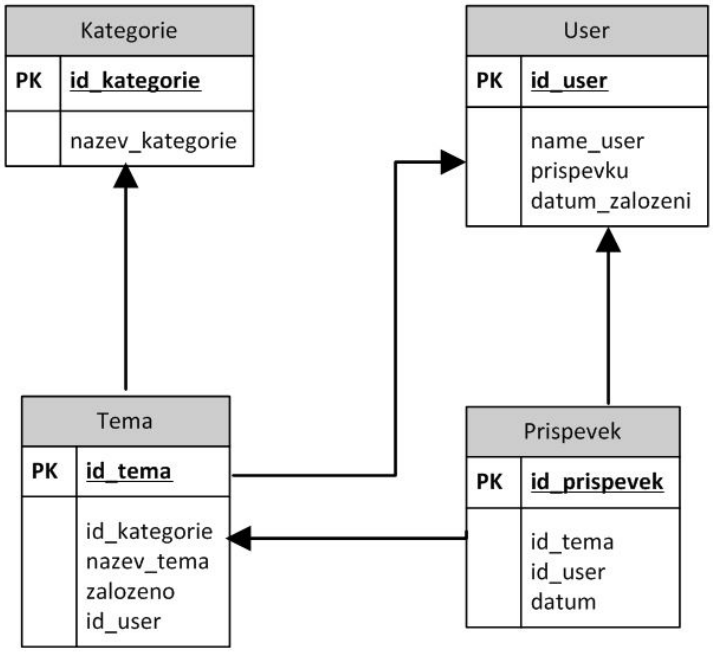
4.3 Databáze

Pro ukládání dat jsme zvolili databázi MySQL v níž jsme vytvořili tabulky Kategorie, Tema, Prispevek a User. Každá tabulka a se skládala z následujících atributů:

- id - jednoznačné identifikační číslo,
- název - popis,

- datum - datum vzniku záznamu.

Blíže je databázové schéma, včetně vztahů, vidět na obrázku 10.



Obrázek 10: Schéma databáze pro uložení dat z internetového fóra

4.4 Program pro extrakci dat

Program pro extrakci dat jsme implementovali v programovacím jazyce Java. Jeho princip je velice jednoduchý. Základem byla dobrá znalost struktury fóra, HTML kódu a regulárních výrazů.

Jak bylo zmíněno výše, každé téma má své unikátní identifikační číslo. Toto číslo je součástí URL adresy jako atribut. Identifikační číslo tématu není nic jiného než pořadové číslo tématu. Tedy první založené téma mělo číslo 1 a nejnovější má už číslo kolem 310 tisíc. Stačilo tedy inkrementovat atribut v URL adrese a získali jsme pro každé téma výpis příspěvků. Jediné co bylo třeba hlídat, byla vícestránková témata. Pokud téma obsahuje více jak 15 příspěvků, je rozděleno na více stránek po 15. Ovšem řešení bylo opět jednoduché. Přidali jsme jednu podmínku, která pomocí regulárního výrazu kontrolovala, zda na konci výpisu příspěvků je odkaz na další stránku. Pokud ano, program dále prohledával příspěvky v tématu, pokud ne, skončil prohledávání a šel na další téma.

Získaný zdrojový kód bylo nutné pomocí regulárních výrazů prohledat a nalézt všechny námi hledané hodnoty týkající se příspěvků. Jedná se o login přispívajícího, datum založení příspěvku, id tématu a id příspěvku. Nalezená data jsme ukládali do předpřipravené databáze.

Po prohledání všech příspěvků bylo nutno doplnit tabulku kategorie. Jednotlivé kategorie s popisem a identifikačním číslem byly podobným způsobem extrahovány z hlavní stránky fóra.

U uživatelů byla situace podobná jako u témat. Každý uživatel má stránku se svým profilem. URL adresa tohoto profilu končí atributem, jehož číslo znamená identifikační číslo uživatele. Toto číslo opět stačilo inkrementovat a získali jsme postupně všechny profily uživatelů na fóru. Pomocí regulárních výrazů jsme extrahovali login, datum založení profilu a počet příspěvků daného uživatele.

4.5 Úprava dat

Surová data získaná extrakcí fóra nejsou vhodná pro funkci shlukovacích algoritmů. Potřebujeme je připravit do podoby vektorového datového modelu.

Pro tyto potřeby jsme zvolili porovnání uživatelů fóra podle příspěvků v jednotlivých kategoriích. Vytvořili jsme v databázi novou tabulku, kde každý uživatel tvoří záznam a jeho atributy jsou počty příspěvků v různých kategoriích fóra. Vznikla tabulka o 22 tisících záznamech a 28 atributech.

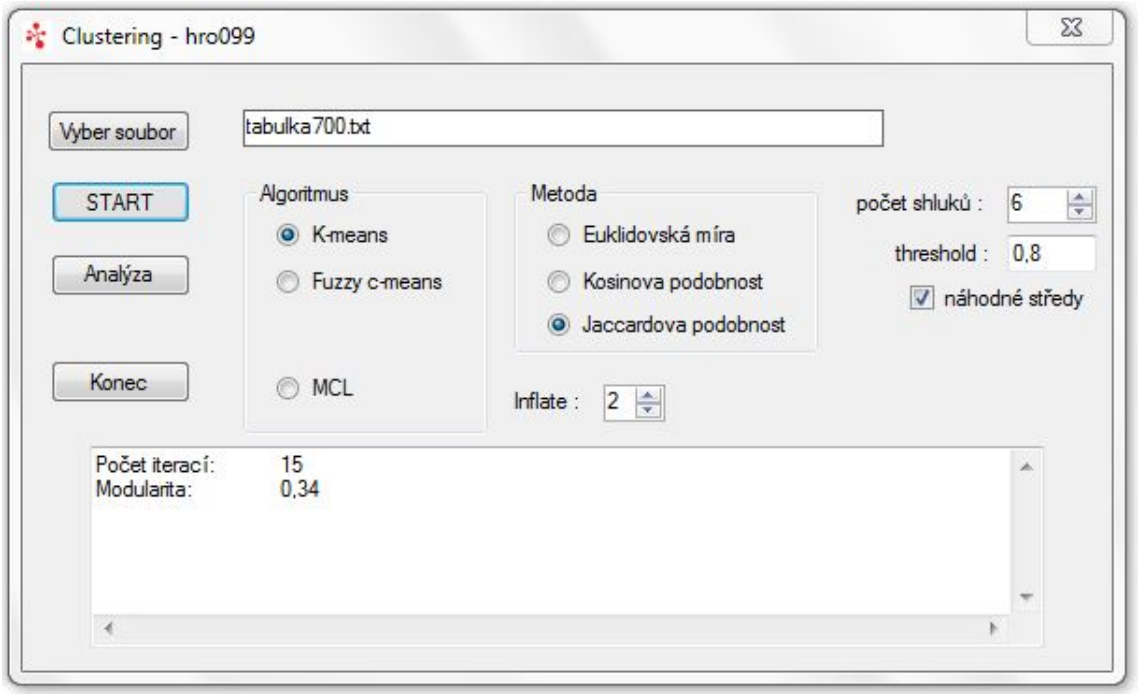
S těmito daty se již dá pracovat, nicméně obsahují velké množství přebytečných údajů. Jedná se o data, kde uživatel nepřispěl do nějaké kategorie žádným příspěvkem. Protože je těchto dat velká většina, můžeme tato data vypustit. Data proto uložíme do textového souboru ve formě řídké matice. Každý řádek představuje jednoho uživatele a data jsou za sebou uloženy v následující podobě „sloupec : počet příspěvků“. Vynecháním nulových hodnot se výrazně sníží velikost dat a algoritmy pak mohou pracovat rychleji.

5 Implementace shlukovacích algoritmů

V následujících podkapitolách si rozebereme postup při implementaci shlukovacích algoritmů. Popíšeme prostředí a funkce, které umožňuje aplikace. Popíšeme také, jaké problémy se vyskytly při implementaci a jaké jsme zvolili řešení. Kapitola obsahuje i diagram tříd, na němž je dobře vidět funkce aplikace a které třídy spolu spolupracují.

5.1 Pracovní prostředí

Pro implementaci shlukovacích algoritmů jsme zvolili prostředí Visual Studio 2010. Jako programovací jazyk byl zvolen $C\#$. Bylo nám jasné, že pro uživatelský komfort a nastavování parametrů shlukovacích algoritmů budeme muset přijít s vhodným grafickým prostředím. Konzolovou aplikaci jsme tudíž zavrhli a vytvořili aplikaci používající rozhraní WindowsForms. Ukázku aplikace si můžete prohlédnout níže na obrázku 11.



Obrázek 11: Ukázka aplikace

5.2 Popis funkcí a možnosti aplikace

Z obrázku 11 je patrné, že aplikace je sice poměrně jednoduchá, nicméně možností k nastavení je zde poměrně hodně. Nejdůležitějším prvkem je samozřejmě možnost výběru shlukovacího algoritmu. Jsou zde na výběr tři možnosti a to K-means, Fuzzy C-means

a Markov Cluster algorithm (MCL). V horní části aplikace je tlačítko na výběr souboru s daty. Volba vhodného datového souboru bude popsána v jiné části této kapitoly. Nevhodně zvolený datový soubor samozřejmě skončí chybovým hlášením.

K vektorovým shlukovacím algoritmům jsou na výběr tři metody pro porovnávání objektů. Jedná se o Euklidovskou míru, kosinovu podobnost a Jaccardovu podobnost. V pravé části aplikace se nastavuje počet shluků a je zde také možnost si vybrat, jestli budou počáteční středy inicializovány náhodně či vylepšenou metodou výběru středů. Dále je zde možnost vyplnit políčko threshold. Threshold je hodnota, kterou už algoritmy nebudou brát v potaz. Jedná se hlavně o vyloučení odlehlých hodnot, které by výrazně zkreslily konečný výsledek. K thresholdu se váže tlačítko Analýza, které spustí analýzu vybraných dat s vybranou metodou. Na výsledném grafu hodnot, které vyjadřují vypočítané míry vzdáleností či podobností mezi objekty, můžeme vyčíst hodnoty odlehlých hran a stanovit threshold. Threshold není povinný parametr, nicméně doporučujeme udělat před spuštěním algoritmů analýzu dat. Získáme tím aspoň zhruba představu o shlukovaných datech.

Grafový shlukovací algoritmus MCL nepotřebuje ke svému správnému chodu žádná složitá nastavení. Je zde možnost jen nastavit parametr inflate, který byl popsán v teoretické části a defaultně je dle doporučení autora algoritmu nastaven na 2.

Ve spodní části aplikace je textové okno, kde se vypisují informace o shlukovacím procesu. Příkladem mohou být hodnoty středů, které se po každé iteraci přepočítávají.

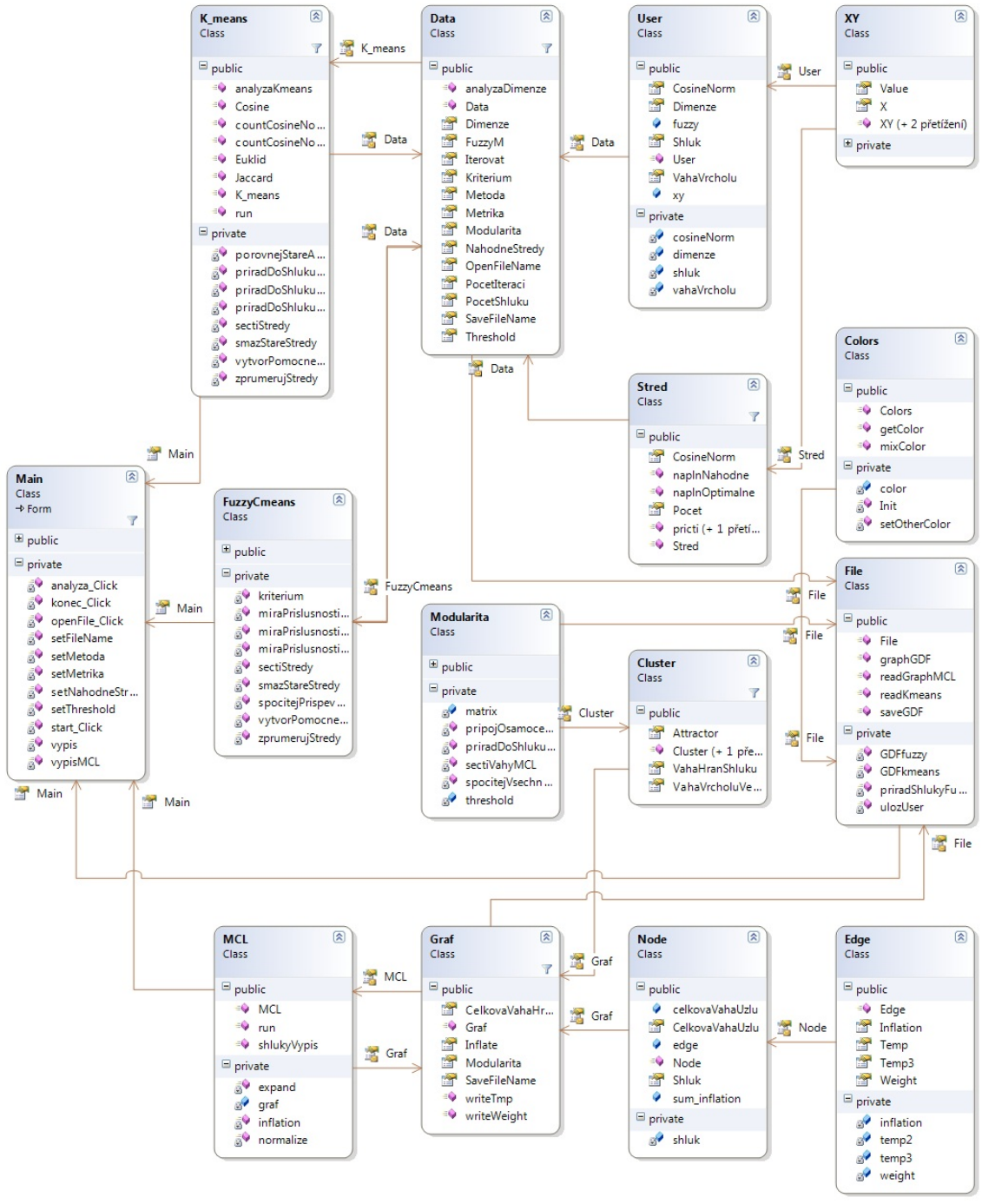
5.3 Diagram tříd

Na obrázku 12 níže si lze prohlédnout diagram tříd. Celá aplikace se skládá z 15 tříd, jež jsou mezi sebou různě provázané. Aplikace vychází z hlavní třídy Main, která tvoří formulář z obrázku 11, a která inicializuje ostatní třídy. Dále je vždy inicializována třída s daty, podle toho, který typ algoritmu je zvolen. Algoritmus MCL používá třídu Graf, kde jsou data uložena ve formě uzlů a hran. K tomuto účelu slouží třídy Node a Edge. Algoritmy K-means a Fuzzy C-means využívají ke svému chodu třídu Data, kde jsou data uložena ve formě řídké matice. Data jsou načtena a uložena pomocí třídy File, která spolupracuje s třídou Color pro výslednou barvu shluků v aplikaci Gephi. Dále je zde třída Stred, která má na starost uchovávání pozic středů při použití vektorových algoritmů. Modularita je počítána ve třídě Modularita, která spolupracuje se třídou Cluster, jež uchovává jednotlivé shluky.

5.4 Testovací data a graf

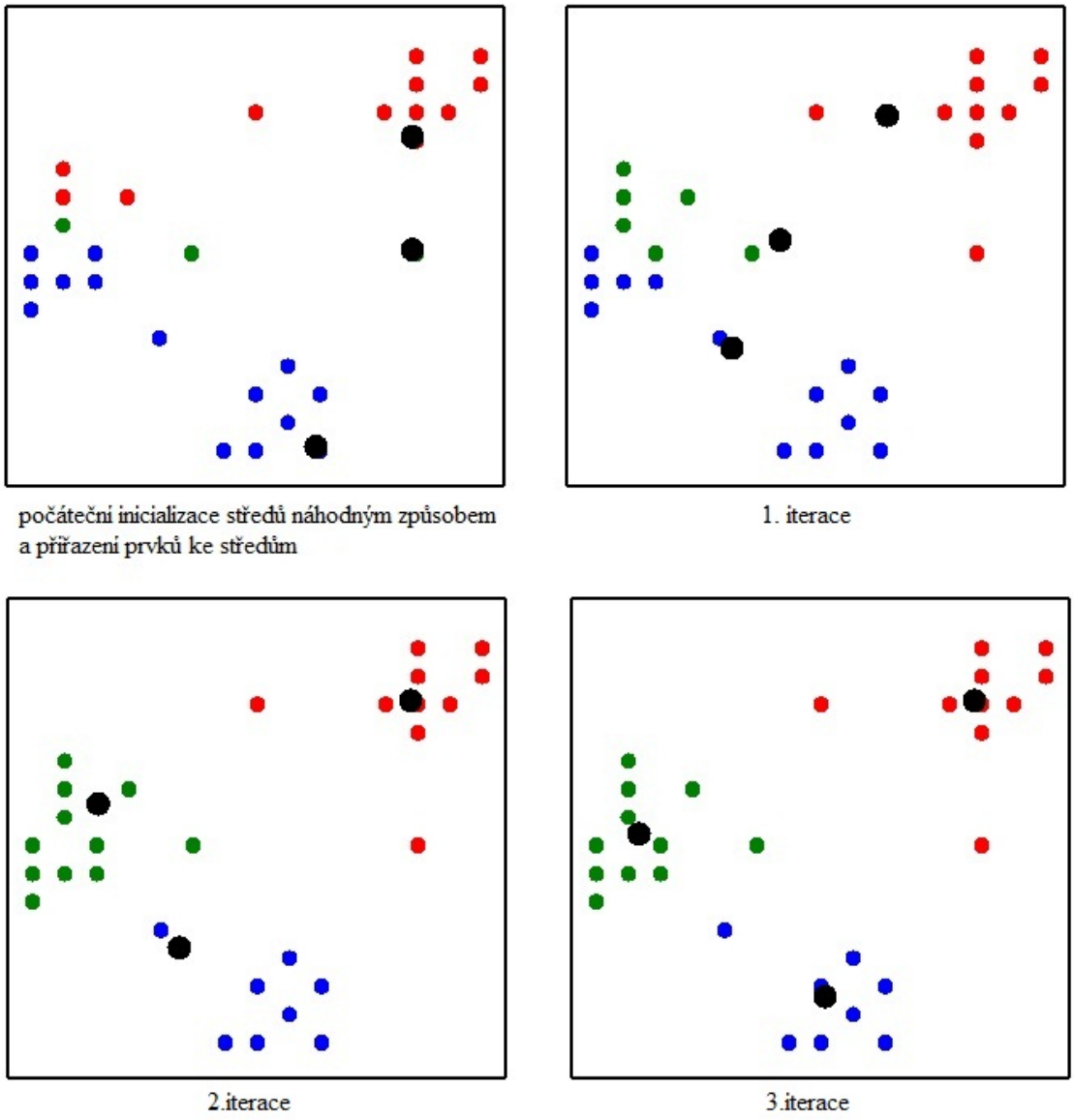
Pro počáteční vývoj aplikace jsme měli připravena testovací data. Jedná se o 30 prvků, jejichž atributy jsou souřadnice X, Y, a jdou jednoduše vykreslit ve dvoudimenzionálním prostoru. Tato data byla záměrně sestavena tak, aby tvořila tři výraznější shluky, a mezi ně jsme dali pár „zbloudilých“ prvků.

Zároveň jsme vytvořili třídu Graf, která tyto prvky vykreslila. Každý prvek má barvu podle toho, ke kterému středu to má nejblíže nebo je mu nejvíce podobný.



Obrázek 12: Diagram tříd

Středy jsou znázorněny velkým černým puntíkem. Na obrázku 13 můžeme vidět celý proces algoritmu K-means. Od počátečního výběru středů, přiřazení do shluků až po přepočítání center shluků.



Obrázek 13: K-means, 3 iterace

5.5 Implementace K-means

Algoritmus K-means pracuje s maticí vzdálenosti, což představuje pro větší množství dat velkou procesorovou zátěž a klade velké paměťové nároky na výbavu počítače. Proto jsme zvolili zápis dat v podobě řídké matice. Řídká matice znamená, že jsou použity jen nenulové hodnoty, které jsou zapsány ve formátu „sloupec:hodnota“. Tyto dvojice hodnot jsou zapsány za sebou a odděleny obyčejnou mezerou. Nový řádek znamená nový prvek a nové atributy. Úspora paměti a následně procesorového času se v případě fóra, které má zhruba 22 tisíc uživatelů, projevila velmi výrazně. Důvodem je, že jsme rozdělili fórum na 28 sekcí (atributů) a zjišťovali, kolik příspěvků měl uživatel v dané sekci. Valná většina uživatelů se spokojila jen s jedním příspěvkem, a když už napsali příspěvků více, tak většinou do stejné kategorie. Proto kdybychom nepoužili zápis dat v řídké matici, obsahoval by zápis zbytečně 27 nul a jen jedno podstatné číslo.

S ohledem na zápis dat v řídké matici jsme museli zvolit i tomu odpovídající datovou strukturu při implementaci. Situaci jsme vyřešili pomocí třídy Data. Třída Data obsahuje kromě jiných pomocných proměnných dva seznamy. Jeden pro seznam všech uživatelů a jeden pro seznam středů. Třída User, definující jednoho uživatele, obsahuje seznam jeho atributů. Třídu atributů jsme nazvali XY, kde X znamená sloupec (kategorii fóra) a Y je počet příspěvků. Tímto je zaručeno, že každý uživatel obsahuje seznam všech svých nenulových hodnot příspěvků.

Pro každý typ metody pro porovnávání objektů jsme museli implementovat funkci, která koresponduje se vzorcem v teoretické části této práce. Což nebylo zcela jednoduché, zejména z toho důvodu, že každý objekt měl jiný počet atributů. Tyto atributy se navíc musely shodovat v kategorii, což taky ne vždy platilo. Proto jsou metody pro výpočet vzdáleností či podobností plné podmínek a optimalizovány na co nejrychlejší průchod kódem.

Podobnostní metody jsme pro správnou funkci algoritmu K-means převáděli na vzdálenost pomocí vztahu $D = 1 - S$. Pro podobnost i vzdálenost platí nepřímá úměra. Čím je podobnost větší, tím větší je shoda mezi objekty a jsou si více podobné. U vzdálenosti je tomu přesně naopak. Menší vzdálenost mezi objekty znamená, že jsou si více podobné.

Celý algoritmus pracuje iterativně v cyklu, kdy se kontroluje, zda přepočítané středy jsou shodné se středy z minulé iterace. Pokud jsou shodné, znamená to, že algoritmus už se nebude dále přepočítávat, středy se už nikam neposunou a může skončit. Pokud nejsou shodné, algoritmus začne další iteraci. Počet iterací lze výrazně snížit vhodným výběrem počátečních středů.

Pro výběr počátečních středů máme k dispozici dvě metody. Jedna, která náhodně vybere objekty, jenž se stanou počátečními středy. Tato metoda je sice rychlá a jednoduchá na implementaci. Nevýhodou je, že může vybrat podobné objekty a tím znehodnotit výsledek shlukování, případně prodloužit dobu výpočtu. Druhá, progresivnější metoda je sice náročnější, ale v konečném důsledku může ušetřit čas a zlepšit kvalitu shluků. Funguje na principu hledání nejvzdálenějších prvků od už vyhledaných středů.

Výstupem analýzy je soubor, který umožní následnou vizualizaci pomocí Gephi. Datový soubor pro Gephi může mít několik formátů. My jsme zvolili formát GDF, který má strukturu podobnou definici grafu. Nejdříve se definují vrcholy. Povinný je pouze název,

barva a případně velikost zobrazeného vrcholu jsou volitelné atributy. Poté následují hrany a jejich váhy. Zde se váhou myslí podobnost mezi objekty. Při použití euklidovské vzdálenosti je potřeba tuto vzdálenost znormovat a převést na podobnost. Normováním je myšleno nalézt maximální hodnotu, kterou následně podělíme ostatní vzdálenosti. Získáme tím číslo v intervalu $< 0, 1 >$, které odečteme od 1 a dostaneme podobnost. U podobnostních metod se hodnoty normovat nemusí. Ukázku jednoduché struktury gdf souboru můžete vidět níže ve výpisu 1.

```
nodedef> name VARCHAR, color VARCHAR, width float
a ,'255,0,0',  4.0
b ,'0,255,0',  5.0
c ,'0,255,0',  6.0
d ,'255,0,0',  4.0
edgedef> node1 VARCHAR, node2 VARCHAR, weight float
a,b,0.5
a,c,1
a,d,0.8
```

Výpis 1: Ukázka GDF souboru

Při výpisu do souboru se zároveň počítá i modularita. Modularita je popsána v kapitole 3 a slouží k získání kvality shlukování. Sčítáme váhu hran uvnitř shluků a celkovou váhu všech hran. Výsledky modularity se potom spolu s počtem iterací vyhodnocují v kapitole s experimenty.

Důležitým prvkem při výpisu dat do souboru je threshold. Jedná se o omezující hodnotu podobnosti, která zaručuje, že nebudou použity hrany s hodnotou nižší, než je threshold. Threshold je zde z toho důvodu, aby se zabránilo přehlcení dat slabými mezishlukovými hranami. Ve shlucích má threshold jen poloviční hodnotu.

5.6 Implementace Fuzzy C-means

Algoritmus Fuzzy C-means vychází z algoritmu K-means, ale oproti němu je robustnější a má více použití. Největší rozdíl je v interpretaci výsledků. K-means přiřazuje objekt právě do jednoho shluku, zatímco Fuzzy C-means udává míru příslušnosti ke každému shluku.

Implementace se od K-means příliš nelišila a dá se říci, že obě třídy jsou dost podobné. Nejdůležitější změny se týkají hodnot, se kterými se pracuje. U K-means jsme používali jen vzdálenost, ale u Fuzzy C-means se navíc počítá i s mírami příslušnosti k jednotlivým shlukům. Míru příslušnosti k jednotlivým shlukům jsme počítali následujícím způsobem. Sečetli jsme vzdálenosti ke středům shluků a následným podělením jsme získali procentuální poměr k danému shluku.

Jelikož algoritmus Fuzzy C-means nemá striktně přiřazené objekty do shluků, nedá se zcela jednoduše spočítat modularita. Nicméně jsme se pokusili aspoň o vizualizaci výsledných shluků v Gephi. V Gephi je možné nastavit vrcholům barvu v modelu RGB. V K-means jsou vrcholy vybarveny podle příslušnosti ke shlukům, ale zde jsme se barvy pokusili namíchat podle procentuální příslušnosti ke shlukům. Představa byla taková,

že vrchol patřící stejným poměrem do dvou shluků získá barvu, jež vzejde smícháním barev obou shluků. Tímto jsme docílili zajímavého mixu barev.

Další věcí, kterou jsme realizovali v rámci algoritmu Fuzzy C-means, byl výpočet počtu příspěvků pro jednotlivé uživatele. Každý uživatel na fóru napsal nějaký příspěvek, někdo více, někdo méně. Aby grafická vizualizace odpovídala ještě více realitě a uživatelé s více příspěvky byli více vidět, implementovali jsme funkci, která poměrově podle počtu příspěvků zvýrazňovala častěji přispívající uživatele. Tato funkce přiřazovala do výpisu pro Gephi číslo, které udává velikost vykresleného vrcholu.

5.7 Implementace Markov Cluster algorithm

Markov Cluster algorithm je shlukovací algoritmus pracující s daty ve formě ohodnoceného grafu. Vstupem je datový soubor, kde na každém novém řádku je hrana, reprezentovaná dvěma vrcholy. Tyto vrcholy jsou odděleny tabulátorem. Za nimi někdy bývá i váha hrany. Grafy se dále dělí na orientované a neorientované. My jsme v našich experimentech používali výhradně neorientované hrany.

Pro datovou strukturu tvořící graf, jsme připravili několik tříd. Hlavní třídou se stala třída Graf, která jak je už z názvu patrné zapouzdřovala celou strukturu grafu. Základem této třídy je slovník (Dictionary) obsahující všechny vrcholy. Slovník je datová kolekce, umožňující efektivním způsobem vyhledávat požadovanou hodnotu. Tvoří ho dvojice „klíč:hodnota“. Klíč je unikátní, což z něj dělá ideálního reprezentanta vrcholů. Hodnotu v našem případě představuje instance třídy Node. Node je třída, která definuje jednotlivé vrcholy a obsahuje pomocné proměnné pro shlukování a výpočet modularity. Ovšem základem třídy Node je opět slovník, ale tentokrát obsahující instanci třídy Edge, definující jednotlivé hrany.

Samotný algoritmus pracuje v několika krocích. Nejdříve přichází na řadu normování připojených vrcholů v matici sousednosti. Normováním myslíme přepočítání vah hran takovým způsobem, že sečteme všechny váhy hran vrcholu a tyto váhy pak podělíme celkovým součtem. Získáme tím normovanou matici sousednosti, které se taky říká stochastická matice.

Dalším krokem algoritmu je tzv. expanze neboli mocnění matice. Tato hodnota je dle doporučení autora defaultně nastavena na $e = 2$. Mezivýsledky uchováváme v pomocných proměnných třídy Edge.

Následuje operace inflace, která má za úkol opět znormovat umocněnou matici. Její princip je jednoduchý. Už umocněnou matici opět umocní, ovšem ne celou, ale jen jednotlivé prvky matice zvlášť. Nastavuje se pomocí parametru inflate v hlavním okně aplikace a většinou se volí hodnoty v rozmezí $< 2, 3 >$. Poté už následuje zmiňované normování matice. Výsledkem této metody je, že silnější hrany budou silnější a slabší hrany budou slábnout.

Kroky expanze a inflace běží v cyklu tak dlouho, dokud matice nezkonverguje. Ke konvergenci se dostaneme většinou již po pár iteracích. Matice se již nadále nemění a pro každý sloupec zůstanou jen nejsilnější hrany, které poté analyzujeme. Prohledáme diagonálu zkonvergované matice a hledáme nenulové prvky, které se stanou tzv. attractory. Attractor znamená, že na sebe váže ostatní hrany a společně tvoří jeden shluk. Postupně tedy pro-

hledáváme matici a přiřazujeme nenulové prvky k jednotlivým attractorům. Vzniklým shlukům poté jen přiřadíme číslo shluku a vypíšeme.

Výpis provádíme opět ve formě GDF souboru pro vizualizační program Gephi.

5.8 Implementace modularity

Součástí vývoje shlukovacích algoritmů je i třída počítající modularitu. Modularita nám udává kvalitu shlukování. Název této třídy je Modularita.

Třída má dva konstruktory, jeden pro K-means a druhý pro MCL. Součástí MCL větve jsou i metody přiřazení do shluků a počítání vah hran. Pro shluky je vytvořena třída Cluster, kterou tvoří proměnná attractor, definující představitele shluku a dále proměnné pro součty vah hran uvnitř shluku nebo součty vah vrcholů.

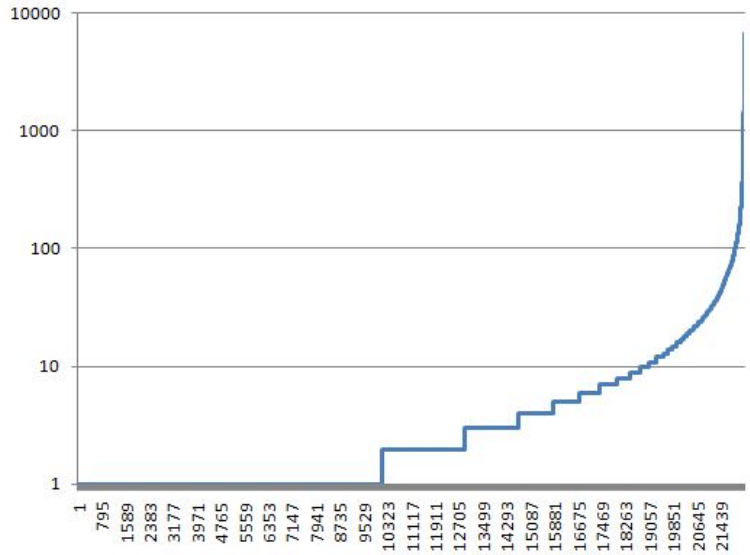
Na závěr je pro každý shluk zvlášť spočítána modularita a jejich součet je uložen do třídy zapouzdřující data, s nimiž se pracovalo.

6 Experimenty a vizualizace

V této kapitole se budeme zabývat experimenty s implementovanými algoritmy. Budeme hodnotit kvalitu shlukování a porovnávat s vizualizačním programem Gephi. Každý experiment je slovně vysvětlen a jsou vyvozeny závěry. Nejdříve v této kapitole provedeme analýzu exportovaného internetového fóra a poté si představíme vizualizační program Gephi. Dále už přijdou na řadu zmiňované experimenty, nejdříve pro K-means, poté představíme grafické vizualizace s Fuzzy C-means a nakonec experimenty pro Markov Cluster algorithm.

6.1 Analýza dat internetového fóra

Internetové fórum notebooky-forum.notebook.cz patří, co se týká velikosti, spíše do průměru. Přesto je na něm registrováno přes 22 tisíc uživatelů, kteří už napsali 310 tisíc příspěvků. Při analýze dat jsme ovšem zjistili, že počet aktivně přispívajících není mnoho. Na obrázku 14 můžeme vidět graf, znázorňující počet příspěvků jednotlivých uživatelů. Na ose X jsou uživatelé a na ose Y počet příspěvků. Osa Y má z důvodu větší přehlednosti zvoleno logaritmické měřítko. Z grafu je patrné, že necelá polovina uživatelů nenapsala více jak jeden příspěvek.



Obrázek 14: Počet příspěvků jednotlivých uživatelů fóra

Do experimentů jsme zahrnuli jen nejaktivnější uživatele z toho důvodu, že málo přispívající uživatelé by byli buď úplně stejní, nebo by tvořili velké samostatné shluky a ovlivnili tím celou analýzu. Do datového souboru jsme vybrali tedy jen uživatele, kteří napsali více jak 50 příspěvků. Bylo to hlavně z důvodu, abychom mohli provést objektivní testy a výsledky porovnat v Gephi, které má jisté paměťové omezení, o kterém se zmiňujeme v podkapitole Gephi níže.

Druhý datový soubor byl tvořen sedmi sty náhodně vybranými uživateli fóra. Počet sedmi set uživatelů byl zvolen proto, protože počet hran, kdy každý uživatel měl vazbu na všechny ostatní uživatele, odpovídal zhruba hranici, kolem které Gephi bylo schopno ještě vrátit nějaké výsledky.

6.2 Gephi

Pro vizualizaci výsledných shluků či vykreslení sítě jsme vybrali open source program Gephi. Tento program je napsán v programovacím jazyce Java a obsahuje spoustu layoutů. Layout je algoritmus pro různé vykreslení grafu. Například pro vizualizaci extrahovaného internetového fóra se nám osvědčil layout s názvem OpenOrd. Tento layout nejvíce odpovídal realitě a pěkně odděloval shluky.

Každý layout obsahuje několik parametrů, které se dají uživatelsky nastavit a ovlivnit tím vykreslení. U layoutu OpenOrd byl nejdůležitějším parametrem Edge Cut, který jak už je z názvu patrné ořezával hrany. Jelikož Gephi pracuje s podobností, která je reprezentována pomocí váhy hran, bylo toto nastavení v intervalu $\langle 0, 1 \rangle$. Gephi nebralo v potaz hrany s menší vahou než nastavený Edge Cut. Defaultně je v Gephi nastavena hodnota 0.8 a tuto hodnotu jsme v rámci objektivity všech experimentů neměnili.

Součástí Gephi jsou i funkce počítající různé statistiky v grafu. Jednou z těchto funkcí je i modularita. Modularita udává kvalitu shlukování tím, že porovnává hustotu hran uvnitř shluků s celkovým počtem hran. V Gephi je tato funkce nastavena na hledání co nejvyšší modularity. Tuto optimální modularitu budeme porovnávat s modularitou, která při shlukování vyšla nám.

Gephi, jak jsme při testech zjistili, má bohužel i svá omezení. Nepodařilo se nám v programu načíst více jak 300 tisíc hran. Když jsme postupovali podle návodu a zvyšovali paměť přidělenou Javě, Gephi nešlo vůbec spustit. Proto jsme se rozhodli přijmout toto omezení a vytvořit testovací data takovým způsobem, aby data neobsahovala více jak 300 tisíc hran. Při experimentech nás zajímalo vykreslení sociální sítě přes layout OpenOrd a výsledek optimální modularity.

6.3 Experimenty s algoritmem K-means

Cílem experimentů bylo prověřit všechny námi naimplementované jednotlivé funkce algoritmu K-means. Jednalo se především o vliv počtu shluků na kvalitu shlukování (modularitu). Dále jsme prověřovali, jak moc se liší náhodný výběr počátečních středů od optimalizovaného výběru. A v neposlední řadě nás zajímal výběr vhodné metody pro porovnání objektů. Byly implementovány tři tyto metody. Jedná se o Euklidovskou vzdálenost, kosinovu podobnost a Jaccardova podobnost. Všechny tři metody jsme zahrnuli do experimentů a každou zvlášť vyhodnotili.

První datová sada pro experimenty s algoritmem K-means se skládala z dat uživatelů fóra, kteří napsali více jak 50 příspěvků. Fórum se skládá z 28 kategorií, do kterých mohou registrovaní uživatelé vkládat své příspěvky. Pro každého uživatele jsme vytvořili tabulku s počtem příspěvků v jednotlivých kategoriích. Pomocí K-means jsme tedy porovnávali jednotlivé uživatele podle toho, do které kategorie na fóru píší příspěvky.

Euklidovská vzdálenost								
	datová sada 1				datová sada 2			
	optimal. středy		náhodné středy		optimal. středy		náhodné středy	
k	iter.	mod.	iter.	mod.	iter.	mod.	iter.	mod.
5	7	0,000	15	0,001	2	0,001	11	0,001
10	9	0,000	17	0,001	9	0,000	12	0,001
15	12	0,001	23	0,001	6	0,001	17	0,000
20	9	0,001	16	0,000	6	0,001	18	0,000
25	13	0,001	21	0,001	7	0,001	18	0,000
30	14	0,000	24	0,000	9	0,000	19	0,000
Kosinova podobnost								
	datová sada 1				datová sada 2			
	optimal. středy		náhodné středy		optimal. středy		náhodné středy	
k	iter.	mod.	iter.	mod.	iter.	mod.	iter.	mod.
5	15	0,137	12	0,117	9	0,289	6	0,302
10	12	0,146	17	0,144	10	0,328	8	0,302
15	6	0,141	11	0,113	11	0,324	8	0,313
20	11	0,119	9	0,095	7	0,320	11	0,301
25	15	0,107	13	0,085	4	0,320	10	0,269
30	13	0,090	9	0,072	4	0,306	8	0,264
Jaccardova podobnost								
	datová sada 1				datová sada 2			
	optimal. středy		náhodné středy		optimal. středy		náhodné středy	
k	iter.	mod.	iter.	mod.	iter.	mod.	iter.	mod.
5	14	0,127	20	0,145	10	0,329	12	0,342
10	17	0,149	21	0,148	13	0,348	10	0,350
15	13	0,138	25	0,133	15	0,335	30	0,348
20	14	0,121	25	0,118	12	0,344	25	0,329
25	24	0,104	34	0108	13	0,344	19	0,306
30	43	0,091	26	0,095	12	0,338	25	0,296
Gephi								
počet shluků	6				8			
modularita	0,167				0,377			

Tabulka 1: Výsledky měření algoritmu K-means pro dvě datové sady

Druhá datová sada se skládala z náhodně vybraných sedmi set uživatelů a testy jsme koncipovali stejně jako u první datové sady.

Tabulka 1 obsahuje všechny námi naměřené výsledky a je rozdělena na dvě části, jež představují dvě datové sady. Každá datová sada dále obsahuje výsledky pro optimalizovaný výběr počátečních středů a druhá polovina je složena z hodnot pro náhodný výběr středů. Pro každý výběr počátečních středů jsme zaznamenávali počet iterací algoritmu K-means a výslednou modularitu. To vše jsme prováděli pro různé počty shluků k, které jsou uvedeny v prvním sloupci.

Vertikálně je tabulka rozdělena na čtyři části, které představují vybrané metody pro porovnání objektů. Nejdříve se jedná o Euklidovskou vzdálenost, následuje kosinova podobnost a Jaccardova podobnost. Tabulka končí naměřenými výsledky z programu Gephi, které by měly představovat optimální shlukování. Výslednou modularitu i počet shluků potom porovnáváme s námi naměřenými výsledky z naší aplikace.

6.4 Vyhodnocení experimentů s algoritmem K-means

Algoritmus K-means je vhodný pro shlukování vektorových dat, avšak je dobré znát jejich strukturu a především vědět, co chceme zjistit. Pokud se předem neudělá analýza dat a nezjistí se přibližný počet shluků, tak se může stát, že K-means bude vracet různorodé výsledky. Jinak se jedná o velmi rychlý algoritmus a s kombinací správné porovnávací metody objektů a analýzy dat vrací kvalitní výsledky odpovídající realitě.

Analýzu výsledků našeho experimentování s algoritmem K-means můžeme rozdělit na tři části podle toho, kterou funkci algoritmu budeme hodnotit.

6.4.1 Dle výběru počátečních středů

Testovali jsme dvě funkce pro výběr počátečních středů. Jedná se o optimalizovaný a náhodný výběr středů. Principem optimalizované funkce je zajistit, aby počáteční středy byly co nejdále od sebe a nestalo se tak, že se zvolí středy blízko u sebe. Slibovali jsme si od něj jednak menší počet iterací algoritmu K-means a poté větší kvalitu shluků tj. vyšší modularitu.

Výsledky ukazují, že naše očekávání byla naplněna a opravdu je počet iterací menší, než u náhodné metody. Co se však příliš nezlepšilo, byla modularita. V průměru se sice modularita u obou testovaných datových sad zlepšila, ale ne nijak výrazně. Co se však díky optimalizované metodě výběru počátečních středů zlepšilo, je stabilita výsledků modularity. U náhodné metody hodnoty výrazně kolísaly, což je zapříčiněno náhodným výběrem středů. Ovšem i tato metoda má svá pozitiva, kde díky náhodnosti dokáže v některých případech vracet lepší výsledky modularity, než optimalizovaná metoda.

6.4.2 Dle zvolené porovnávací metody

Námi zvolená testovací data se neukázala býti vhodná pro Euklidovskou vzdálenost. Při experimentech jsme zjistili, že některé vzdálenosti mezi objekty byly tak velké, že většinou vznikl v algoritmu K-means jeden velký shluk, který si k sobě přitáhl většinu objektů

a ty nejvzdálenější prvky se staly samostatnými shluky. Tento jev nás vedl k zamyšlení a následnému prověření implementace výpočtu vzdálenosti, avšak nebyla zjištěna žádná chyba. Z toho taky vychází naše závěrečné zhodnocení, že Euklidovská vzdálenost není vhodnou mírou pro tento typ dat.

Opačná situace panovala u podobností. Obě podobnosti dávaly lepší výsledky než Euklidovská vzdálenost, avšak Jaccardova podobnost se ukázala být nejlepší volbou pro naše datové kolekce. Už pro malé počty shluků dosahovala dobrých výsledků modularity a v porovnání s Gephi příliš nezaostávala. Jednotlivé objekty (uživatelé fóra) byly rovnoměrně rozděleny do několika shluků a nenastala situace jako u Euklidovské vzdálenosti, kdy se utvořil jeden obrovský shluk a kolem něj pár malých shluků.

Kosinova podobnost se také neukázala být špatnou porovnávací metodou. Výsledky jsou sice o něco horší než v případě Jaccardovy podobnosti, ovšem u první testovací sady dat vyšly výsledky téměř stejně.

6.4.3 Dle počtu shluků

Počet shluků, který je vstupem algoritmu K-means má velký vliv na výslednou modularitu. Podle Gephi měla mít první testovací datová kolekce 6 shluků a nízkou modularitu 0,167. Druhá datová sada, která byla tvořena 700 náhodně vybranými uživateli fóra, by měla mít 8 shluků a modularitu 0,377.

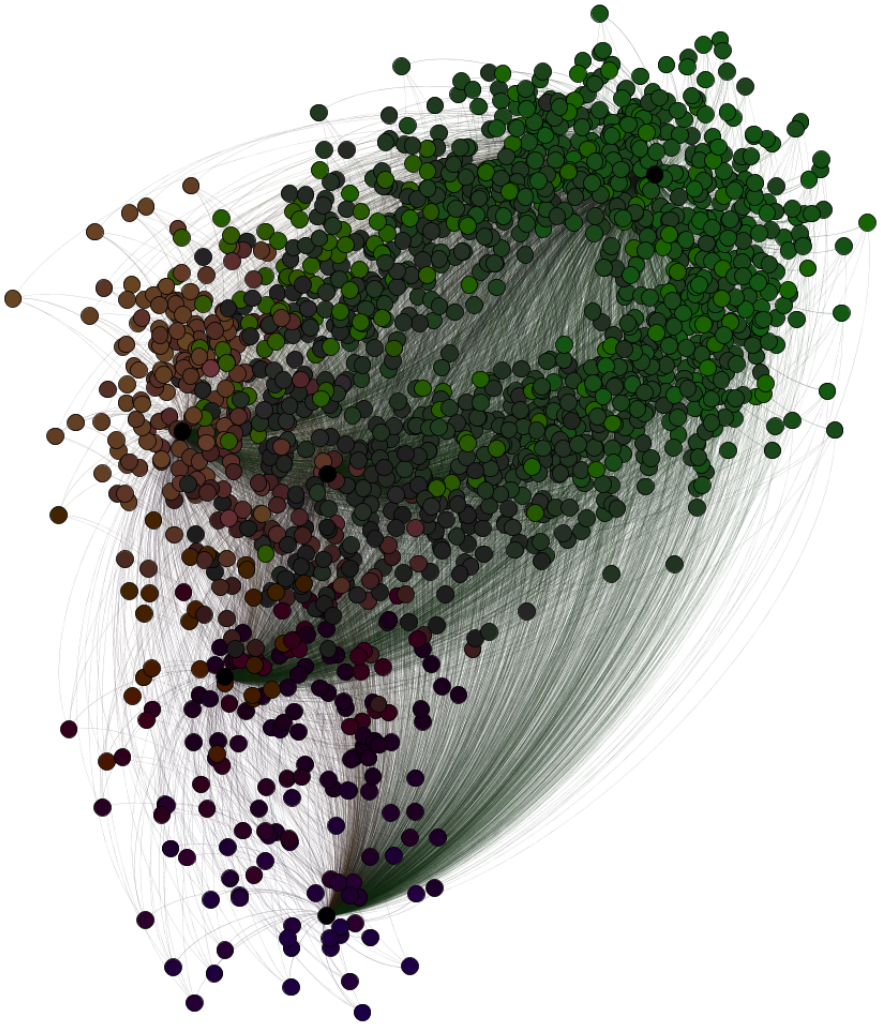
Námi naměřené výsledky tomu odpovídají a modularita je nejvyšší právě kolem těchto počtu shluků. Potvrdili jsme si správnost implementace algoritmu K-means a mohli sledovat, jak se zvyšujícím se počtem shluků se snižuje modularita, což značí, že klesá kvalita shlukování.

6.5 Shlukovací algoritmus Fuzzy C-means

Jelikož algoritmus Fuzzy C-means nepřiřazuje prvky do shluků striktně, ale poměrově, nemohli jsme provést test modularity. Proto jsme se rozhodli, že algoritmus zkusíme vylepšit alespoň po vizuální stránce. Výsledkem je graf, kde jednotlivé shluky jsou odděleny barevně a zároveň čím jsou prvky dál od středu, tak jejich barva slábne a míchá se s barvou druhého nejbližšího shluku.

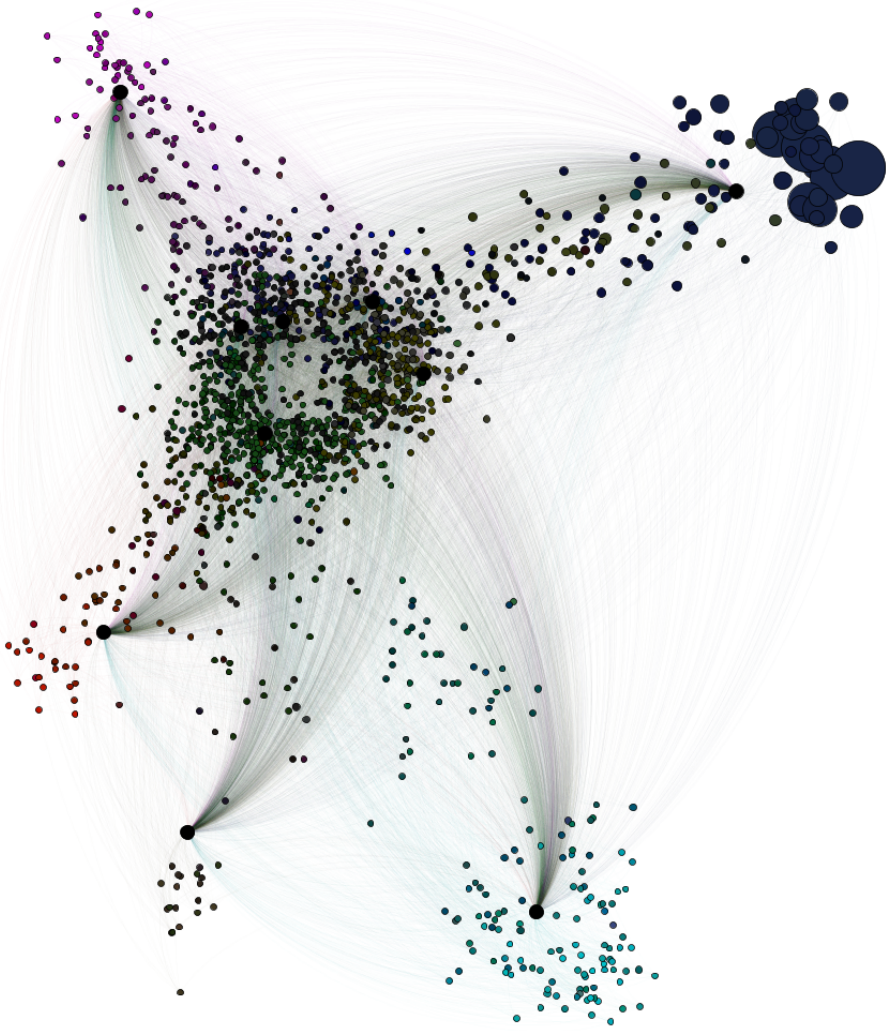
Tohoto efektu jsme docílili pomocí poměrového míchání barev, kdy jsou barvy zadány pomocí RGB modelu a přiřazeny k jednotlivým shlukům. Každý shluk má tedy svou barvu. U každého objektu nalezneme dvě nejvyšší příslušnosti ke shlukům a těmito příslušnostmi vynásobíme jednotlivé složky RGB, které poté sečteme a zprůměrujeme.

Výsledek si můžete prohlédnout na obrázku 15. Data byla použita stejná jako v případě K-means, kdy byli porovnáváni uživatelé fóra dle toho, do které kategorie psali příspěvky. Bylo zvoleno 5 shluků a byla použita Jaccardova podobnost. Výsledkem je jeden velký zelený shluk, který na levé straně postupně přechází k dalším shlukům. Námi implementované zobrazení není úplně ideální, jelikož nemáme možnost v Gephi určit, jak se má výsledná síť zobrazit. Nicméně je zde vidět jisté prolínání mezi shluky a z obrázku lze vyčíst, které prvky inklinují k jakým shlukům.



Obrázek 15: Fuzzy C-means zobrazení příslušnosti ke shlukům

Další funkcí, kterou jsme vylepšili v rámci vizualizace Fuzzy C-means, bylo zvýraznění uživatelů podle počtu napsaných příspěvků. Toto zvýraznění je provedeno pomocí zvětšujícího se kolečka, znázorňující daného uživatele. Pěkně je tato situace vidět na obrázku 16 níže, kde fialový shluk na pravé straně obsahuje právě tyto uživatele. Jedná se o shluk převážně administrátorů nebo moderátorů fóra, kteří píší hodně příspěvků. V experimentu bylo zvoleno 10 shluků a z obrázku je patrné, že toto číslo není optimální, jelikož se uprostřed utvořil jeden velký shluk, který se ovšem skládá z pěti menších shluků. Z toho vyplývá, že optimální počet shluků pro tato data je šest.



Obrázek 16: Fuzzy C-means zvýraznění uživatelů

6.6 Shlukovací algoritmus MCL

Cílem experimentů s algoritmem MCL bylo prověřit jeho funkcionalitu nad různými datovými kolekcemi. Na internetu jsme objevili různé datové kolekce, z nichž jsme vybrali čtyři. Liší se od sebe svou strukturou, množstvím uzlů a hran. Pro každou datovou sadu jsme provedli shlukování algoritmem MCL, výsledky porovnali s Gephi a provedli závěrečnou analýzu sítě.

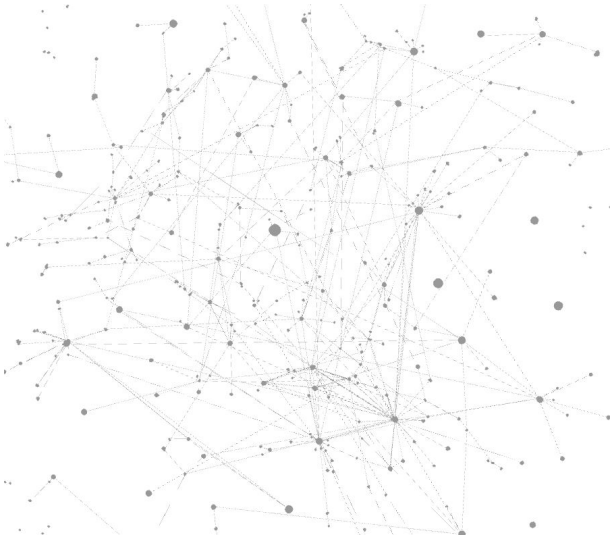
6.6.1 Datová sada - EVA

EVA je výzkumný projekt, jehož název je vytvořen zkratkou anglických slov Extraction, Visualization & Analysis. Tento projekt kombinuje získávání informací, vizualizaci a analýzu sociálních sítí. Cílem je zmapovat vztahy mezi podniky. Zaměřují se na majetkové vztahy mezi firmami a snaží se identifikovat nejvlivnější firmy.

Pro naši analýzu jsme získali data tvořená 7 253 uzly a počet hran byl roven 6 711. Naměřené hodnoty si můžete prohlédnout v tabulce 2 níže.

	MCL	
	modularita	shluků
inflate = 2	0,89	1 189
inflate = 3	0,88	1 198
Gephi	0,957	759

Tabulka 2: Datová sada - EVA



Obrázek 17: Výsledný graf dat EVA

Z obrázku 17 je patrné, že síť je velice řídká a je tvořená velkým počtem malých shluků, které jsou mezi sebou propojeny. To nám ostatně potvrzují i naměřené hodnoty. Naměřili jsme zde největší modularitu ze všech měření. Modularita by měla být největší při malém počtu shluků. Zároveň by však tyto shluky měly mít co největší počet hran uvnitř sebe a malý počet hran mezi sebou. Jelikož je tato datová sada velice řídká, kdy je počet hran větší než počet uzlů, výsledkem je velké množství malých osamocených shluků a vysoká modularita.

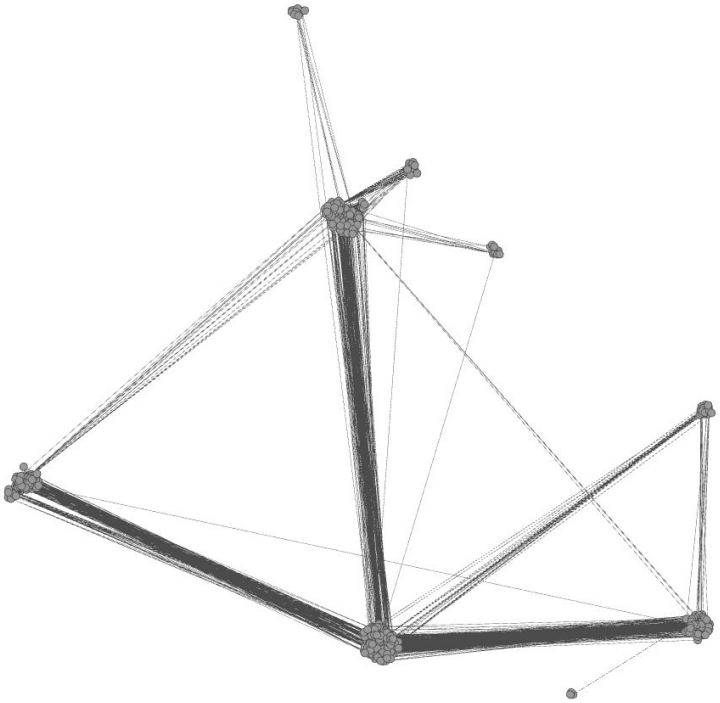
Výsledkem analýzy by mohl být závěr, že firmy jsou ve styku jen se svými nejbližšími partnery. Firem, které spolupracují s větším množstvím podniků je velmi málo.

6.6.2 Datová sada - Facebook

Podařilo se nám získat data i z největší sociální sítě současnosti, z Facebooku. Jedná se o zpracovaná data mezi 747 lidmi, kdy hrana mezi lidmi představuje přátelství.

	MCL	
	modularita	shluků
inflate = 2	0,46	28
inflate = 3	0,412	34
Gephi	0,529	7

Tabulka 3: Datová sada - Facebook



Obrázek 18: Výsledný graf dat Facebook

Výsledky shlukování jsou poměrně zajímavé, můžete si je prohlédnout v tabulce 3. Přestože graf obsahuje malý počet uzlů, množství hran je veliké, až 30 tisíc. MCL shlukování vykazuje oproti optimálnímu Gephi vyšší granularitu shluků, proto i modularita vyšla menší. Zvýšený inflační parametr způsobil, že se síť ještě více rozdělila na menší části.

Na grafu 18 jde krásně vidět jednotlivé shluky lidí, kteří se navzájem znají a přátelí se. Občas se najde pár lidí z dané komunity, kteří znají někoho z jiné komunity. Na tomhle principu funguje známé doporučování přátel, se kterými byste se mohli seznámit.

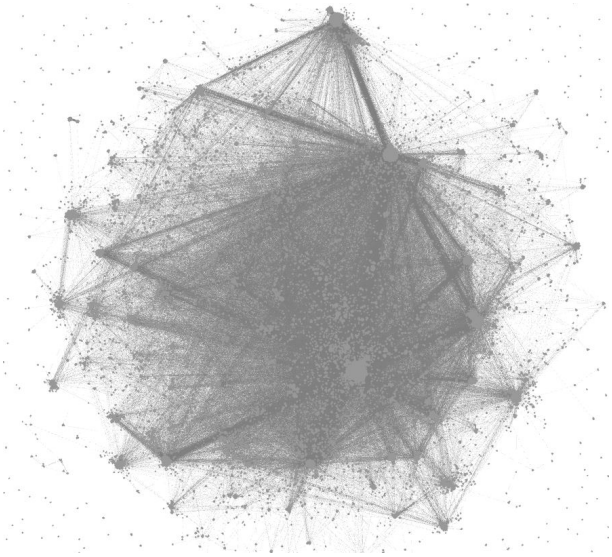
6.6.3 Datová sada - Brightkite

Brightkite byla online sociální síť, jež měla za cíl zprostředkovat svým uživatelům možnost vyhledání blízkých přátel pomocí svého aktuálního umístění. Uživatel pomocí mobilní aplikace zadal svou polohu a na mapce mohl vidět, kde se nachází jeho přátelé. Zobrazovali se ale třeba i lidé, kteří jsou součástí sítě a se kterými má možnost se seznámit. Tato sociální síť v dnešní době již nefunguje.

Získali jsme datovou sadu, kterou tvoří 58 228 uzlů a 214 078 hran. Jedná se o data znázorňující přátelství mezi uživateli, kdy hrana mezi vrcholy znamená, že se oba uživatelé znají a mají se uložené v seznamu přátel.

	MCL	
	modularita	shluků
inflate = 2	0,289	14 502
inflate = 3	0,264	14 876
Gephi	0,678	710

Tabulka 4: Datová sada - Brightkite



Obrázek 19: Výsledný graf dat Brightkite

Z výsledků v tabulce 4 je patrné, že algoritmus MCL síť rozdělil na obrovské množství shluků. Postupně se ukazuje, že pokud síť netvoří pevnou strukturu uvnitř shluků, postará se shlukovací algoritmus o její rozdělení do více komunit.

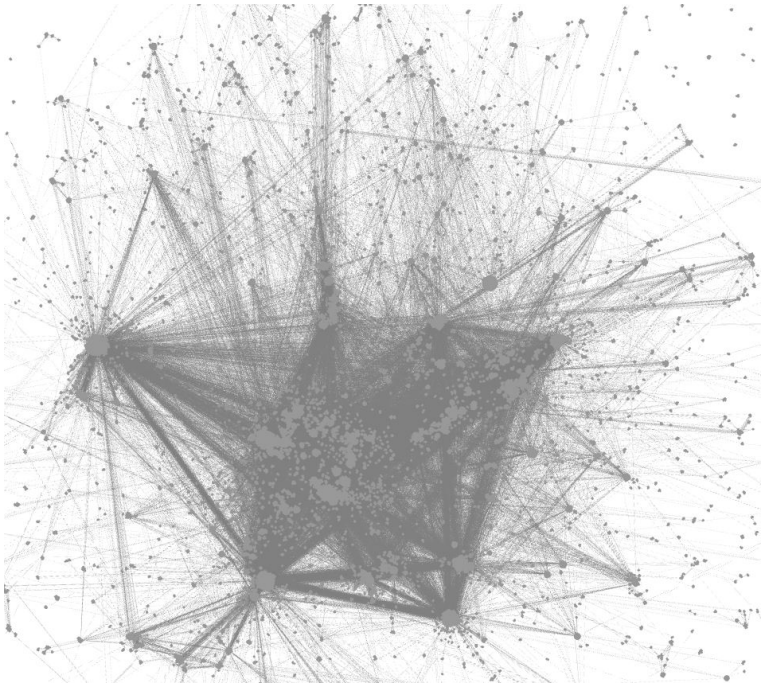
Síť je tvořena několika velkými shluky, jenž jsou propojeny s velkým množstvím menších shluků. Celkově celá síť působí hodně propojeně, ale množství shluků je zde opravdu veliké. Výslednou síť si lze prohlédnout na obrázku 19.

6.6.4 Datová sada - Email-Enron

Jedná se o e-mailovou komunikaci mezi zaměstnanci společnosti Enron. Enron byla americká energetická společnost, které se dařilo, a byla dávána za vzor do doby, než se našly chyby v účetnictví. V roce 2002 musela firma vyhlásit bankrot. Tato datová sada vznikla později ze zveřejněných informací z vyšetřování FBI.

	MCL	
	modularita	shluků
inflate = 2	0,40	2 766
inflate = 3	0,377	2 843
Gephi	0,602	1328

Tabulka 5: Datová sada - Email-Enron



Obrázek 20: Výsledný graf dat Enron

Výsledky jsou o něco lepší než v minulém případě, ale stále se nám zdá, že algoritmus síť rozdělí až moc na malé části. Výsledky modularity 5 stále pokulhávají za optimem z Gephi.

Vizuálně jde podle obrázku 20 vidět lepší strukturu uvnitř shluků, což odpovídá i menšímu počtu shluků proti Brightkite síti.

6.7 Zhodnocení algoritmu MCL

Algoritmus MCL se choval přesně podle popisu autora tohoto algoritmu. Jeho největším problém je přílišná granularita shluků. Pokud struktura v grafu tvořící shluk není dostatečné propojená, algoritmus ji rozdělí. Tato vlastnost se nejvíce projevila u sociální sítě Brightkite, kde algoritmus vytvořil 14 500 shluků a Gephi jen 700. U sítí typu Facebook, kde shluky byly jednoznačné, si algoritmus vedl naopak velice dobře.

Doporučení platí stejné jako u K-means. Je dobré předem vědět, s jakými daty pracujeme, podrobit je analýze a případně i vykreslit v nějakém vizualizačním programu. Potom si lépe dokážeme představit, proč vyšly takové výsledky, jaké vyšly.

7 Závěr

Dnešní svět je neuvěřitelně propojený a sociální sítě se stávají běžnou součástí života. Stejně jako když přišel první mobilní telefon, každý si ho chtěl hned vyzkoušet. Pro všechny to byl malý zázrak, telefonovat bez drátů. Postupem času se to stalo normální a dnes už si bez něj nikdo nedovede život představit. Stejný osud čeká i sociální sítě, které se již dostaly do širokého povědomí lidí a stala se z nich „normální“ věc.

S rostoucím počtem lidí, kteří se stávají součástí různých sociálních sítí, roste i jejich vliv. Dnes už existují specializované algoritmy, které prohledávají sociální sítě a sbírají data o uživatelích. Tato data jsou následně analyzována a dají se z nich poté s nějakou pravděpodobností určit například pohyby kurzů na burze, či kdy dojde k nepokojům v některé zemi. Analýza dat je mocná věc a firmy si za ni nechávají dobře zaplatit.

Jedním z mnoha odvětví, které se zajímají o analýzu dat ze sociálních sítí, je marketing. Pomocí sběru dat si o Vás zjistí co nejvíce informací a už jste chyceni. Najednou na Vás při prohlížení internetu začne vyskakovat reklama s produktem a Vy si říkáte, jak to jen vědí, že zrovna toto potřebujete. Říká se tomu cílená reklama a přijdou na to pomocí analýzy dat.

Součástí analýzy dat, kromě statistických metod, mohou být například i shlukovací algoritmy. Tyto algoritmy se snaží v datech najít podobné struktury či vlastnosti. Těmto podobným strukturám říkáme shluky. Tato diplomová práce se zabývá třemi shlukovacími algoritmy. Každý z nich vrací odlišné výsledky a každý má jiné využití. Jedná se o shlukovací algoritmus K-means, Fuzzy C-means a MCL neboli Markov Cluster algorithm.

Každá analýza začíná sběrem dat. Proto jsme implementovali program, který měl za úkol zpracovat data z předem vybraného fóra. Pro tyto účely jsme vybrali internetové fórum notebooky-forum.notebook.cz, které se zabývá notebooky. Jedná se o docela živé fórum a dat bylo k dispozici více než dost. Našim cílem bylo analyzovat uživatele fóra a vzájemně je porovnat podle toho, do které z kategorií na fóru píší příspěvky. Implementace byla poměrně jednoduchá a data jsme získali za pár hodin.

Daleko těžší byla implementace shlukovacích algoritmů. Při programování nastaly nečekané problémy, které nás ze začátku vůbec nenapadly. Jedná se zejména o práci s velkými datovými kolekcemi. Zjistili jsme, že paměť není nekonečná a že má své limity. To nás vedlo k hlubšímu zamyšlení a posléze k myšlence uchovávání dat v podobě řídké matice, kdy se vypouští nulové hodnoty.

Na závěr naší práce jsme se věnovali experimentům se shlukovacími algoritmy. Pro algoritmus MCL jsme z internetu stáhli několik odlišných datových sad. Mírným zklamáním bylo, že algoritmus MCL až příliš prořezává graf a výsledkem je pak velké množství shluků. Výsledky jsme porovnávali s vizualizačním programem Gephi, který se nakonec stal i součástí analýzy. Kvalitu shlukování jsme měřili pomocí modularity a porovnávali s modularitou v Gephi.

Pro K-means jsme použili naše data získaná extrakcí diskuzního fóra. Testovali jsme mnoho věcí. Od vlivu výběru počátečních středů až po výběr vhodné metody pro porovnání objektů. Z experimentů jsme zjistili, která metoda je vhodná pro určitý druh dat a která naopak nevhodná. Zklamáním byla metoda počítající Euklidovskou vzdále-

nost, která vykazovala hodně špatné výsledky. Naopak vynikajícími výsledky se může pochlubit Jaccardova podobnost, která se v některých případech rovnala i Gephi.

S algoritmem Fuzzy C-means jsme prováděli spíše jen vizuální úpravy a snažili se o co nejlepší vykreslení překrývajících se shluků. Výsledek jsme testovali v Gephi a můžete si jej prohlédnout v kapitole s experimenty.

Po vyhodnocení všech experimentů jsme došli k závěru, že každý algoritmus se hodí na něco jiného. Každý zpracovává jiná data a je na uživateli, aby si rozmyslel, co vlastně chce získat. Z našeho pohledu je velmi důležitá analýza dat před samotným shlukováním. Zjistit si o datech co nejvíce, pak porovnávat s výsledky a vyhodnocovat. Nelze k datům jen tak přijít a začít shlukovat. Proto doporučujeme se vždy důkladně připravit.

Celkově se nám práce na toto téma velmi líbila, a kdybychom měli možnost vybrat si ji znovu, tak neváháme. Přínosem je jednak zlepšení našich programátorských dovedností, ale hlavně prohloubené znalosti o analýze dat.

8 Reference

[1] A. Abraham, A.E. Hassanien, and V. Snášel. *Computational Social Network Analysis: Trends, Tools and Research Advances*. Computer communications and networks. Springer London, 2010.

[2] C.C. Aggarwal. *Social network data analytics*. Springer US, 2011.

[3] Takao Asano, Shin-Ichi Nakano, Yoshio Okamoto, and Osamu Watanabe, editors. *Algorithms and Computation - 22nd International Symposium, ISAAC 2011, Yokohama, Japan, December 5-8, 2011. Proceedings*, volume 7074 of *Lecture Notes in Computer Science*. Springer, 2011.

[4] A.L. Barabási. *V pavučině sítí*. Fénix (Prague, Czech Republic). Paseka, 2005.

[5] L. Buštíková. Analýza sociálních sítí. *Sociologický časopis*, (15):193–206, 1999.

[6] B.Y. Cao, C. Zhang, and T. Li. *Fuzzy Information and Engineering*. Number sv. 1 in Advances in soft computing. Springer Berlin Heidelberg, 2009.

[7] Carlos Castillo. *Effective Web Crawling*. PhD thesis, University of Chile, 2004.

[8] Steven J. Cox. Cell Assemblies: A Binary Integer Programming Problem. [online]. [cit. 2013-04-16]. Dostupné z: `http://cnx.org/content/m30936/latest/?collection=col10523/latest`, 31.8.2009.

[9] L. da F. Costa, A. Evuskoff, G. Mangioni, and R. Menezes. *Complex Networks: Second International Workshop, CompleNet 2010, Rio de Janeiro, Brazil, October 13-15, 2010, Revised Selected Papers*. Communications in Computer and Information Science. Springer, 2012.

[10] S. M. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, The Netherlands, 2000.

[11] S. Feldt, P. Bonifazi, and R. Cossart. Dissecting functional connectivity of neuronal microcircuits: experimental and theoretical insights. *Trends Neurosci*, 34(5):225–36, 2011.

[12] G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM series on statistics and applied probability. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2007.

[13] Brian Halabisky. Measuring Dis/Similarities Between Objects (Cells) In 'n'-Dimensional Space. [online]. [cit. 2013-04-21]. Dostupné z: `http://tonto.stanford.edu/~brian/making_measurements.html`, 29.6.2009.

[14] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering algorithms and validity measures. In *Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on*, pages 3 –22, 2001.

[15] D. Hansen, B. Shneiderman, and M.A. Smith. *Analyzing Social Media Networks with NodeXL: Insights from a Connected World*. Elsevier Science, 2010.

[16] M. Hazewinkel. *Encyclopaedia of Mathematics (9)*. Encyclopaedia of Mathematics. Springer, 1993.

[17] P. Hebák and et al. *Vícerozměrné statistické metody 3.* Informatorium, Praha, 2005.

[18] D.E. Holmes and L.C. Jain. *Data Mining: Foundations and Intelligent Paradigms: Volume 3: Medical, Health, Social, Biological and Other Applications*. Data Mining: Foundations and Intelligent Paradigms. Springer, 2012.

[19] J. Jandourek. *Sociologický slovník*. Portál, 2001.

[20] David Knoke and Song Yang. *Social network analysis*, volume 154 of *Quantitative applications in the social sciences*. Sage, Los Angeles, CA, 2nd ed edition, 2008.

[21] Ted G. Lewis. *Network Science: Theory and Applications*. Wiley Publishing, 2009.

[22] Bao-Liang Lu, Liqing Zhang, and James T. Kwok, editors. *Neural Information Processing - 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part III*, volume 7064 of *Lecture Notes in Computer Science*. Springer, 2011.

[23] Kathy Macropol. Clustering on graphs: The markov cluster algorithm (mcl). CS Department of Computer Science, University of California, Santa Barbara, 2009.

[24] M. Meloun, J. Militký, and M. Hil. *Statistická analýza vícerozměrných dat v příkladech*. Academia, Praha, 2012.

[25] Vincent Moulton and Mona Singh, editors. *Algorithms in Bioinformatics, 10th International Workshop, WABI 2010, Liverpool, UK, September 6-8, 2010. Proceedings*, volume 6293 of *Lecture Notes in Computer Science*. Springer, 2010.

[26] Alexandru Nedelcu. Data Mining: Finding Similar Items and Users. [online]. [cit. 2013-04-21]. Dostupné z: `https://www.bionicspirit.com/blog/2012/01/16/cosine-similarity-euclidean-distance.html`, 16.1.2012.

[27] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, 2003.

[28] M. E. J. Newman. *Networks: an introduction*. Oxford University Press, Oxford; New York, 2010.

[29] Ondřej Platko. Sociální sítě 1.díl. [online]. [cit. 2013-04-15]. Dostupné z: `http://owebu.bloger.cz/Internet/Socialni-site-1-dil`, 29.6.2009.

[30] J. Prokop. *Algoritmy v jazyku C a C++ - 2.* Průvodce (Grada). Grada, 2012.

[31] F. Rennie and T. Morrison. *e-Learning and Social Networking Handbook: Resources for Higher Education*. Taylor & Francis, 2012.

[32] H. Řezanková, D. Húsek, and V. Snášel. *Shluková analýza dat*. Professional Publishing, 2009.

[33] Dylan Stewart. Internet Communities And Forums. [online]. [cit. 2013-04-08]. Dostupné z: `http://www.videojug.com/interview/internet-communities-and-forums-2#what-is-an-internet-forum`, 31.3.2008.

[34] Julie Teninbaum. The Business of Facebook. [online]. [cit. 2013-04-10]. Dostupné z: `http://www.fastcompany.com/1740204/business-facebook`, 25.4.2011.

[35] L. Wang. *Fuzzy Systems and Knowledge Discovery[: Third International Conference, FSKD 2006, Xián, China, September 24-28, 2006 : Proceedings*. Lecture notes in artificial intelligence. Springer-Verlag New York Incorporated, 2006.

[36] D.E. Wittkower. *Facebook and Philosophy: What's on Your Mind?* Popular Culture and Philosophy Series. Open Court Publishing Company, 2010.

[37] X. Yan. *Ictis 2011: Multimodal Approach to Sustained Transportation System Development*. American Society of Civil Engineers, 2011.